

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

A DIGITAL FILTER REPRESENTATION OF THE ASQ-81
MAGNETOMETER

by

Michael Charles Huete

September 1983

Thesis Advisor: Andrew R. Ochadlick

Approved for public release, distribution unlimited

T210142

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Digital Filter Representation of the ASQ-81 Magnetometer		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis September 1983
7. AUTHOR(s) Michael Charles Huete		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, Ca. 93943		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, Ca. 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE September 1983
		13. NUMBER OF PAGES 121
		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release, distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Digital Filter, AN/ASQ-81, Magnetometer, Magnetic Anomaly Detection, MAD, Bilinear Transformation, FORTRAN, Geomagnetic Noise, Geomagnetic Field		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A digital filter representation of the ASQ-81 magnetometer is derived from the s-plane transfer functions of the system through the use of a bilinear trans- formation. A FORTRAN computer program is written which applies this represen- tation to time-sampled total magnetic field data in order to obtain a time series representation of ASQ-81 filtered total field. A series of simulations and a field experiment are conducted which verify the program output. Appli- cations of this program include usage in conjunction with geomagnetic field		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

data in order to produce a new data set representative of geomagnetic noise observed by Navy MAD (Magnetic Anomaly Detection) aircraft with the potential to investigate techniques of reducing geomagnetic noise in MAD aircraft.

S/N 0102- LF- 014- 6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Approved for public release, distribution unlimited

A Digital Filter Representation of the ASQ-81
Magnetometer

by

Michael Charles Huete

Lieutenant, United States Navy

B. S. E. E., Tulane University, 1976

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY
(Antisubmarine Warfare)

from the

NAVAL POSTGRADUATE SCHOOL

September 1983

ABSTRACT

A digital filter representation of the ASQ-81 magnetometer is derived from the s-plane transfer functions of the system through the use of a bilinear transformation. A FORTRAN computer program is written which applies this representation to time-sampled total magnetic field data in order to obtain a time series representation of ASQ-81 filtered total field. A series of simulations and a field experiment are conducted which verify the program output. Applications of this program include usage in conjunction with geomagnetic field data in order to produce a new data set representative of geomagnetic noise observed by Navy MAD (Magnetic Anomaly Detection) aircraft with the potential to investigate techniques of reducing geomagnetic noise in MAD aircraft.

TABLE OF CONTENTS

I.	INTRODUCTION	9
II.	GEOMAGNETICS REVIEW	11
A.	EARTH'S MAGNETIC FIELD	11
1.	Constituents of the Geomagnetic Field	11
2.	Elements of the Magnetic Field Vector	13
III.	ASQ-81 MAGNETOMETER	15
A.	DESCRIPTION OF SYSTEM OPERATION	15
B.	TRANSFER FUNCTIONS	18
IV.	DIGITAL FILTER MODELLING OF SYSTEMS	20
A.	SEQUENCE REPRESENTATION OF TIME FUNCTIONS	20
1.	Signal Representations	20
2.	Linear Shift-Invariant Systems	21
B.	THE z -TRANSFORM	24
1.	Description of the z -Transform	24
2.	The Bilinear Transformation	25
C.	THE DIGITAL COMPUTATIONAL ALGORITHM	26
D.	THE CASCADE FORM OF THE COMPUTATIONAL ALGORITHM	28
V.	EXPERIMENTAL PROCEDURE	31
A.	SIMULATION	31
1.	Sinusoidal Inputs	31
2.	Noiselike Inputs	36

3.	Anderson Function Simulations . . .	40
4.	Impulse Function Response	44
B.	EQUIPMENT SETUP	46
VI.	CONCLUSIONS	59
APPENDIX A:	AN/ASQ-81 FILTER TRANSFER FUNCTIONS . . .	62
APPENDIX B:	AN/ASQ-81 z-TRANSFORM SYSTEM TRANSFER FUNCTIONS FOR DIRECT FORM REALIZATION . .	64
APPENDIX C:	AN/ASQ-81 z-TRANSFORM SYSTEM TRANSFER FUNCTIONS DIRECT FORM II REALIZATION . .	67
APPENDIX D:	z-TRANSFORM REALIZATION DIFFERENCE EQUATIONS	68
APPENDIX E:	DIGITAL SOFTWARE FOR SIMULATION - DIRECT FORM (SINUSOIDS AS INPUT)	71
APPENDIX F:	DIGITAL SOFTWARE FOR SIMULATION - CASCADE FORM (SINUSOIDS AS INPUT)	76
APPENDIX G:	DIGITAL SOFTWARE FOR COMPUTATION OF SYSTEM AMPLITUDE VERSUS FREQUENCY	81
APPENDIX H:	DIGITAL SOFTWARE FOR SIMULATION (ANDERSON FUNCTIONS AS INPUT)	86
APPENDIX I:	DIGITAL FILTERING SOFTWARE	91
	LIST OF REFERENCES	118
	BIBLIOGRAPHY	119
	INITIAL DISTRIBUTION LIST	120

LIST OF FIGURES

2.1	Magnetic Field Elements	14
3.1	Metastable Helium Magnetometer	16
3.2	Signal Flow Diagram for ASQ-81	19
4.1	Representation of Transformation of an Input Sequence to an Output Sequence	22
4.2	Cascade Structure With a Direct Form II Realization of Each Second Order Subsystem . .	30
5.1	Output of First Stage Filter of Digital Filter Computer Program With Sinusoidal Input in Simulation	32
5.2	Output of Second Stage Filter Design With Input of Sinusoid	32
5.3	Input Signal to Digital Filter Program. A Sinusoid of Frequency 0.1 HZ and Amplitude ± 1	34
5.4	Output of First Stage of Digital Filter Program	35
5.5	Output of Second Stage of Digital Filter Program	35
5.6	Output of Third Stage of Digital Filter Program	36
5.7	Input to Filter - 0.1 HZ Sinusoid, 10 HZ Sinusoid, Uniformly Distributed Random Noise of Amplitude ± 1	37
5.8	Output of First Filter Stage	37
5.9	Output of Second Filter Stage	38
5.10	Final Filter Stage Output	38
5.11	Plot of Attenuation versus Frequency for Sinusoidal Inputs for Digital Filter and Analog Filter	39
5.12	First Anderson Function Input. CPA at Time 2.5 Minutes	42

5.13	Filter Output for First Anderson Function Input	42
5.14	Second Anderson Function Input. CPA at Time 2.5 Minutes	43
5.15	Filter Output for Second Anderson Function . .	43
5.16	Third Anderson Function Input. CPA at Time 2.5 Minutes	44
5.17	Filter Output for Third Anderson Function . . .	44
5.18	Impulse Response of Filters	45
5.19	Schonstedt Coil Time Series Output	48
5.20	Program Time Series Output With Schonstedt Coil as Input	49
5.21	ASQ-81 Time Series Output	50
5.22	Raw Coil Antenna Time Series Output	54
5.23	Coil Antenna Difference Field Time Series Output	55
5.24	Program Time Series Output With Coil Antenna Difference Field Time Series as Input	56

I. INTRODUCTION

The detection and location of submarines (and other magnetic bodies) through the discrimination of changes or anomalies in the Earth's magnetic field is called Magnetic Anomaly Detection or MAD. In this technique, a magnetometer measures the magnitude of the Earth's magnetic field and provides an indication of that magnitude, or, more usually, an indication of changes in the magnitude of the Earth's field. These changes, or anomalies, can indicate the presence of magnetized bodies which may or may not be a submarine.

The magnetometer currently in use in the United States Navy for use in this MAD process is the AN/ASQ-81 metastable helium vapor total field magnetometer.

Research is currently being conducted at the Naval Postgraduate School in Monterey, California, in various aspects of the applications of magnetometers, including Magnetic Anomaly Detection (MAD). Within the context of this research, magnetic field measurements are made through the use of sets of wire wound coils vice any specific magnetometer or magnetic detecting system. The data collected through the use of these coils is evaluated and

processed in a variety of methods for different project goals.

This thesis project is designed to produce an acceptable alternative to the physical presence of an experimental AN/ASQ-81 magnetometer at the postgraduate school by allowing the determination, in conjunction with other research in progress, of the output of the AN/ASQ-81 magnetometer from the data collected from the school's measurement coils. It is hoped that this will assist future research projects as, for example, in allowing a determination of environmental noise of such characteristics as to affect the AN/ASQ-81 magnetometer operationally with the eventual goal of providing an environmental noise index or a system of removing such noise from the magnetometer-detection system.

II. GEOMAGNETICS REVIEW

A. EARTH'S MAGNETIC FIELD

1. Constituents of the Geomagnetic Field

The most common method of specifying the constituting parts of the geomagnetic field is to divide the field in terms of distance from the center of the Earth. This method results in three classifications: internal, crustal, and external. [Ref. 1]

The internal field originates in the core region and is the most stable field, containing only extremely low frequency temporal variations. The crustal, or anomalous, field arises from modifications made on the internal field by materials and structures in the Earth's crust. These variations are not constant with regard to spatial locations, and comprise part of what is known as geological variations. The external field is the most dynamic and arises from many sources, including the interaction between the solar wind and the Earth's magnetic field.

In addition to this method of defining the Earth's magnetic field is the method of time variations. This method consists of considering that part of the field which varies with periodicities greater than about one year as the

steady field and everything else as the variation field.

[Ref. 2]

The steady field consists of the internal field, also referred to as the main field. Slow variations of the main field with periods of years or longer are referred to as secular variations.

There are various elements that contribute to the geomagnetic field, some of which are external to the Earth's surface. External contributions make up only a small part of the steady field, but play an important role in the variation field. These external sources include current systems in the Earth's upper atmosphere affected by solar electromagnetic radiation and gravitation, solar corpuscular radiation and the interaction of solar plasma with the main field, and the effect of the solar interplanetary field.

[Ref. 3]

The geomagnetic field changes with time. As previously mentioned, very slow variations in the main field with periods of on the order of years to thousands of years are referred to as secular variations. Secular variations are caused by a variation in the strength or orientation of the Earth's center dipole.

Other time variations of the field can be categorized into quiet variation fields and disturbed variation fields. Disturbed variation fields include geomagnetic micropulsations, which are of particular interest to

operational forces as these can mask target signatures and are therefore a source of noise to MAD sensors.

Quiet variation fields are those which are not due to disturbances in the interplanetary environment and which vary slowly and regularly. [Ref. 3]

Disturbed variation fields are geomagnetic field variations that appear to be the result of interplanetary environmental changes and do not possess a simple periodicity. These variations include ionospheric disturbances, the aurora, geomagnetic storms, and geomagnetic micropulsations.

2. Elements of the Magnetic Field Vector

The geomagnetic field vector is characterized at any point by its direction and magnitude. This is commonly accomplished through a system of coordinates as shown in Figure 2.1. The field is measured in terms of local coordinates with respect to true North. [Ref. 3]

The various coordinates are referred to as magnetic elements and are defined as follows:

B: Total field intensity (the symbol F is sometimes also used, as in this figure.)

H: Horizontal component

X: Northward, or NorthSouth component

Y: Eastward, or EastWest component

Z: Downward, or Vertical component

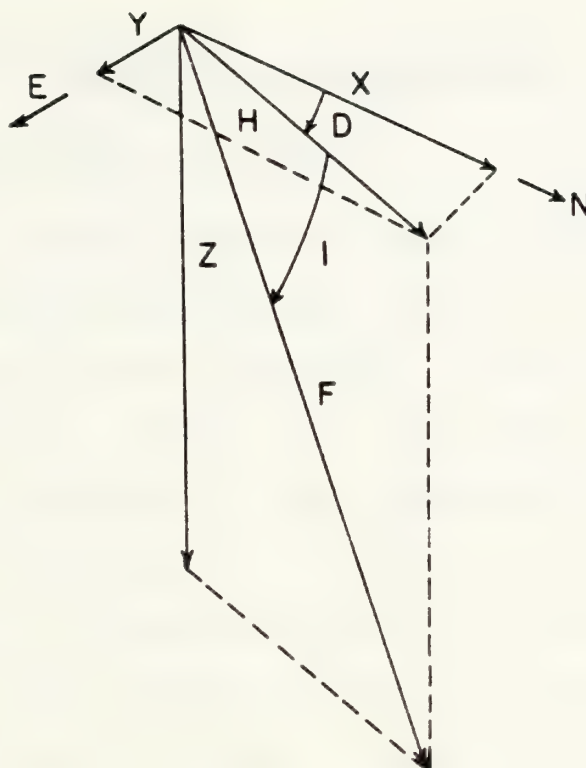


Figure 2.1 Magnetic Field Elements [Ref. 4].

D: Declination or magnetic variation

This is the angle between X and H and
is measured positive eastward.

I Inclination or Dip Angle.

This is the angle between H and B (or F)
and is measured positive downward.

III. THE AN/ASQ-81 MAGNETOMETER

A. DESCRIPTION OF SYSTEM OPERATION

The Magnetic Anomaly Detecting set currently in use in the U S Navy is the AN/ASQ-81 magnetometer. This set is used to locate and classify submerged submarines by sensing disturbances in the Earth's magnetic field (anomalies) caused by the presence of the magnetic mass of the submarine. The disturbance of the Earth's field is detected by the magnetometer, processed through filtering circuits, and amplified. The output signal of the magnetometer is displayed on a chart recorder for interpretation by an operator.

The magnetic detecting set is a metastable helium vapor magnetometer. The operation of the magnetometer is based on the light absorbtion properties of helium gas subjected to certain light stimulus (optical pumping), radio frequency excitation, and the Earth's magnetic field. The magnetometer consists of a helium lamp, lens and polarizer to generate a beam of polarized light radiation. This focused and polarized light beam is directed through a helium absorbtion cell to an infrared (IR) detector. Some of the helium gas in the absorbtion cell is maintained in a metastable state by application of VHF excitation.

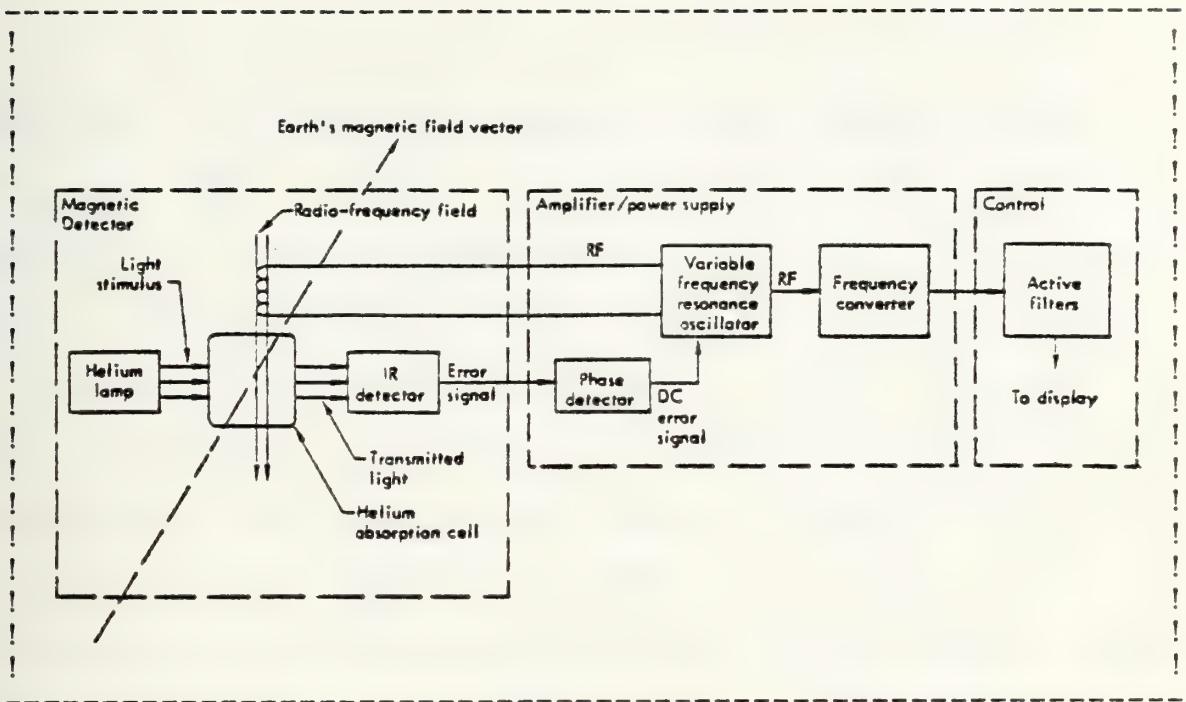


Figure 3.1 : Metastable Helium Magnetometer [Ref.5].

The Earth's magnetic field imposes a magnetic force upon the excited helium vapor atoms to force the atoms into one of three energy sublevels. This is called the Zeeman effect. The rate or frequency of atomic precession caused by this effect is called the Larmor frequency. A helium lamp is used to optically pump the atoms in the absorption cell, with the result that the polarized light energy passing through the absorption cell will polarize (magnetize) the helium atoms in the absorption cells by selectively pumping the Zeeman levels of the energy of the helium atoms in the cell. The magnetization direction is determined by the polarization of the photons from the helium lamp.

RF energy is then introduced to the absorption cell in the form of an additional magnetic field imposed through the use of coils oriented perpendicular to the precessed polarized helium atoms in the absorption cell and energized by a variable frequency RF oscillator. The RF oscillator is tuned to the Larmor frequency, which results in depolarization of the atoms. The atoms attempt to equally repopulate the Zeeman energy levels. However, the helium lamp is still beaming polarized light energy into the absorption cell, causing the atoms to absorb light energy and rise to an excited energy level. This absorption of light energy is detected through the use of an infrared detector. The RF oscillator frequency producing maximum light absorption is called the resonant frequency, and is determined through the use of a servo loop from the infrared detector to the RF variable frequency oscillator.

Therefore, any change in the Earth's magnetic field intensity will result in a change in the Larmor frequency of the helium atoms in the helium absorption cell. This new Larmor frequency will be detected by the ASQ-81 magnetometer. Since the gyromagnetic ratio of helium is 28.024 HZ per gamma, this detection of the resonant frequency provides a measurement of the Earth's magnetic field intensity at any given time. A change in the Earth's

magnetic field intensity could signal the presence of a submerged submarine.

The output resonant frequency developed by the magnetometer is converted to a proportional output voltage which is filtered through the Magnetic Anomaly Detection (MAD) bandpass filters for environmental noise reduction and utilized to drive a chart recorder for observation by an operator. [Ref. 6]

B. TRANSFER FUNCTIONS

Transfer functions for the AN/ASQ-81 filters were obtained from the manufacturer of the AN/ASQ-81 detecting set, Texas Instruments of Dallas, Texas. These transfer functions are listed in Appendix A and are in the form of $H(s)$, that is, the frequency domain, or S domain, where $S = j\omega$. The s -domain representation for transfer functions is routinely utilized to express output system characteristics for given system inputs. As the S domain representation is not utilized further in this discussion, it will not be further explained.

As the output signal of the ASQ-81 magnetometer is filtered through a fixed high-pass system, then through a selectable low pass system and a selectable high pass system (as shown in Figure 3.2 below), the transfer functions are listed in this order.

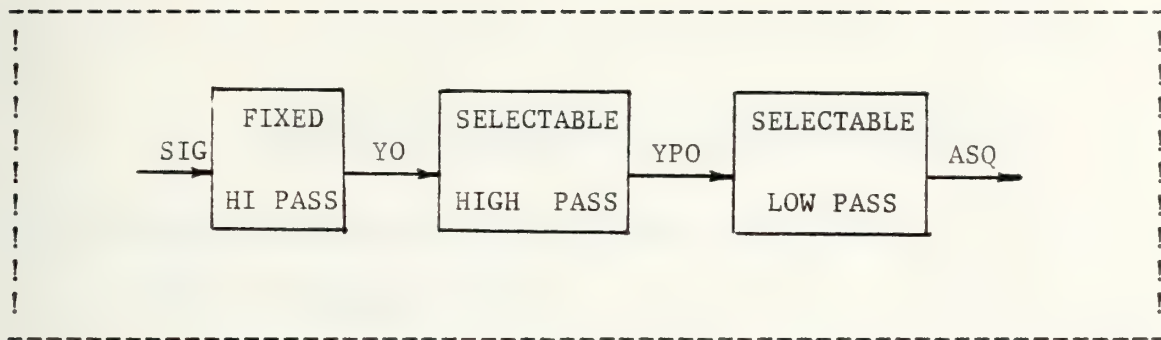


Figure 3.2 : Signal Flow Diagram for ASQ-81

Further discussion will be made of the selectable filters later.

IV. DIGITAL FILTERING MODELLING OF SYSTEMS

A. SEQUENCE REPRESENTATION OF TIME FUNCTIONS

1. Signal Representation

A signal can be defined as a function that conveys information, generally about the state or behavior of a physical system. Although signals can be represented in many ways, the information conveyed by the signal is contained in a pattern of variations of some form. Signals are represented mathematically as functions of one or more independent variables, one of the most common of which is time.

The independent variable of the mathematical representation of a signal may be continuous or discrete. Continuous time signals are signals that are defined over continually values of time and are therefore represented by continuous-variabled functions. Discrete time signals are defined at discrete time intervals and are therefore represented by functions whose independent variable(s) take on discrete values only. Discrete-time signals are represented as sequences of numbers. [Ref.. 7]

In addition to the fact that the independent variables can be either continuous or discrete, the signal amplitude can be either continuous or discrete. Digital

signals are those for which both time and amplitude are discrete. Analog signals are those for which both time and amplitude are continuous.

Digital signal processing deals with transformations of signals that are discrete in both time and amplitude, usually represented by sequences of numbers. The nth number in the sequence x being processed is usually represented as x(n), and is formally written as:

$$x=[x(n)], \quad -\infty < n < +\infty$$

In general, an arbitrary sequence can be expressed as

$$x(n) = \sum_{k=-\infty}^{\infty} x(k) d(n-k)$$

where $d(n-k)$ is the unit sample at time k . In other words, an arbitrary sequence may be expressed as a sum of scaled, shifted unit samples, where the scaling factor is equal to the amplitude of the sequence at that time.

2. Linear Shift-Invariant Systems

A system is defined mathematically as a unique transformation or operator that maps an input sequence $[x(n)]$ into an output sequence $[y(n)]$. This is denoted as:

$$y(n) = T[x(n)]$$

and is often depicted as in Figure 4.1.

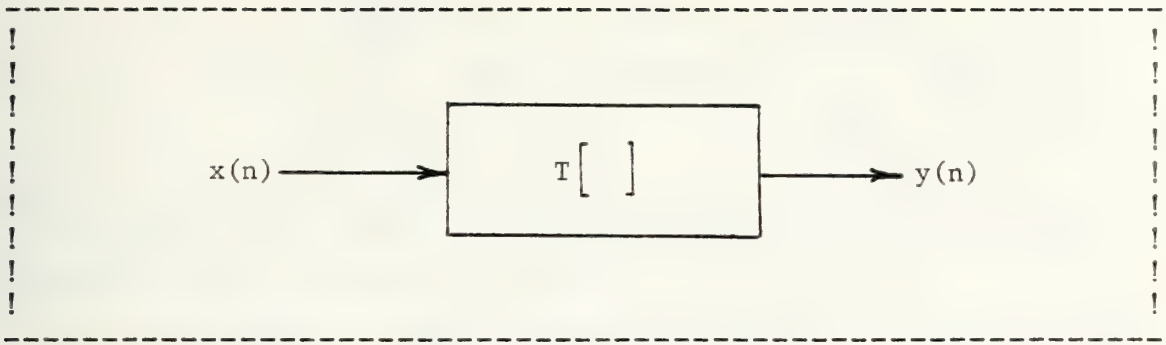


Figure 4.1: Representation of Transformation of an Input Sequence to an Output Sequence. [Ref.7]

Classes of discrete time systems are defined by placing constraints on the transformation $T[\]$.

The class of linear systems is defined by the principle of superposition. If $y_1(n)$ and $y_2(n)$ are the responses when $x_1(n)$ and $x_2(n)$ are the inputs, then a system is linear if

$$\begin{aligned} T[ax_1(n) + bx_2(n)] &= aT[x_1(n)] + bT[x_2(n)] \\ &= ay_1(n) + by_2(n) \end{aligned}$$

for any arbitrary constants a and b . This, together with the concept of representing a sequence by a sum of delayed and scaled unit-sample sequences, suggests that a linear system can be characterized by its unit-sample response. Specifically, let $h_k(n)$ be the response of the system to $d(n-k)$, a unit sample occurring at $n=k$. Then

$$y(n) = T\left[\sum_{k=-\infty}^{\infty} x(k) d(n-k)\right] \quad \text{or,}$$

$$y(n) = \sum_{k=-\infty}^{\infty} x(k) T[d(n-k)] = \sum_{k=-\infty}^{\infty} x(k) h_k(n)$$

Thus the system response can be expressed in terms of the response of the system to $d(n-k)$.

The class of shift invariant systems is characterized by the property that if $y(n)$ is the response to $x(n)$, then $y(n-k)$ is the response to $x(n-k)$, where k is a positive or negative integer. When the index n is associated with time, shift-invariance corresponds to time-invariance. The property of shift invariance implies that if $h(n)$ is the response to $d(n)$, then the response to $d(n-k)$ is simply $h(n-k)$. Therefore

$$y(n) = \sum_{k=-\infty}^{\infty} x(k) h(n-k)$$

and any linear shift-invariant system is completely characterized by its unit-sample response $h(n)$.

A subclass of linear shift-invariant systems consists of those systems for which the input $x(n)$ and the output $y(n)$ satisfy an N th-order linear constant-coefficient difference equation of the form

$$\sum_{k=0}^N a_k y(n-k) = \sum_{r=0}^M b_r x(n-r)$$

If the assumption of causality is made about the system, a linear difference equation provides an explicit relationship between the input to the system and the output of the system. This can be seen by rewriting the previous equation as

$$y(n) = \sum_{k=1}^N c_k y(n-k) + \sum_{r=0}^M d_r x(n-r)$$

where $c_k = -a_k / a_0$ and $d_r = b_r / a_0$.

Thus the n th value of the output can be computed from the n th value of the input and the N and M past values of the output and input, respectively. The difference equation not only represents the system for theoretical purposes, but it may also serve as a computational realization of the system. The z -Transform makes use of this property to realize systems.

B. THE z -TRANSFORM

1. Description of the z -Transform

The z -transform plays an important role in the analysis and representation of discrete-time linear shift-invariant systems. The z -transform, $X(z)$, of a sequence $x(n)$ is defined as

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n}$$

where z is a complex variable. This representation of the z -transform is referred to as the two-sided z transform. The one sided z -transform consists of the same summation for terms of n greater than or equal to zero. For the case that $x(n)=0$ for $n<0$, the one sided and two sided z transforms are equivalent.

By expressing the complex variable z in polar form as $z = re^{j\omega}$, the z -transform can be interpreted as the Fourier transform of $x(n)$ multiplied by an exponential sequence. For $r = 1$, that is, for $|z| = 1$, the z -transform is equal to the Fourier transform of the sequence.

2. The Bilinear Transformation

The transfer functions of analog systems are most often expressed in terms of $s = j\omega$ (see section III B.). This corresponds to the analog frequency response of the system. This analog frequency response can be "mapped", that is, transformed to the z -plane from the s -plane through the use of the bilinear transformation. The effect of utilizing the bilinear transformation is to convert a system transfer function in terms of the variable S into the system transfer function in terms of the variable z . The transformation itself is:

$$s = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}$$

and

$$z = \frac{(2/T) + s}{(2/T) - s}$$

where T is the sampling period, that is, the time between data samples.

Thus a transform can be made from one plane to the other. In this manner, the transfer function, $H(z)$, of a system may be obtained.

The bilinear transformation equations may be shown to hold in general, and the use of this transformation may be shown to yield stable digital filters from stable analog filters [Ref. 7]. The bilinear transformation maps the imaginary $j\omega$ axis in the s -plane onto a unit circle (of the region of convergence) in the z -plane, with the left half s -plane mapped onto the region inside the circle and the right hand (region of instability) s -plane mapped onto the region outside this circle [Ref. 8]. A complete discussion of the z -transform is available in several texts, some of which are listed in the Bibliography.

C. THE DIGITAL COMPUTATIONAL ALGORITHM

In implementing a digital filter on a digital computer such as the IBM 3033, the input-output relationship of the signals through the system being synthesized must be converted to a computational algorithm. The algorithm is specified in terms of a set of basic computations of elements. For the implementation of discrete-time systems

described by linear constant coefficient difference equations, such as the AN/ASQ-81, it is convenient to choose as these elements the basic operations of addition, delay, and multiplication by a constant. The computational algorithm for implementing the filter is then defined by a structure or network consisting of an interconnection of these basic operations. For a system transfer function of the form

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 - \sum_{k=1}^N a_k z^{-k}} = \frac{Y(z)}{X(z)}$$

the difference equation relating input and output is easily written down directly from the system function and is given by

$$y(n) = \sum_{k=1}^N a_k y(n-k) + \sum_{k=0}^M b_k x(n-k) \quad [\text{Ref. 7}]$$

This difference equation can be interpreted directly as a computational algorithm in which the delayed values of the input are multiplied by the coefficients b_k , the delayed values of the output are multiplied by the coefficients a_k , and the resulting products are added. It is now easy to see the process to be followed in obtaining the computational algorithm for the AN/ASQ-81 magnetometer

transfer function. The z-transform of the system transfer function is obtained through the use of the bilinear transformation, and is then converted into a difference equation relating input and output signals, thence to a FORTRAN computer program. A table of z-transforms of system functions is included in Appendix B.

In the FORTRAN computer program realization of the total system computational algorithm, each filter block is transformed into a separate difference equation and algorithm. This was done to enable a "building block" type approach to the program, and to minimize computational and roundoff errors.

D. THE CASCADE FORM OF THE COMPUTATIONAL ALGORITHM

Even though the direct form realization of the digital filter design may be perfectly satisfactory in a theoretical sense, it may be less than desirable in the context of realization through the use of a general purpose computer of fixed register length. The parameters of a digital filter are usually obtained with a high degree of accuracy, which results in a faithful realization of the desired system. When these parameters are quantized, as in a finite memory register within a computer, the frequency response of the resulting digital filter may differ appreciably from the original design. In fact, the quantized filter may fail to

meet design specifications although the unquantized filter does. [Ref. 7]

The sensitivity of the filter response to errors in the filter parameters is dependent upon the structure of the filter realization. Therefore, in the event of an unacceptable change in the frequency response of the filter due to quantization errors, it is often possible to minimize the effect of these errors through an alternate filter realization structure. An alternate structure to the previously discussed direct form realization is the cascade form realization.

The direct form network structures were obtained directly from the system function $H(z)$ written in the form of a ratio of sums. If this ratio is factored into a product of polynomials of the form

$$H(z) = A \prod_{k=1}^{[(N+1)/2]} \frac{1 + B_{1k} z^{-1} + B_{2k} z^{-2}}{1 - a_{1k} z^{-1} - a_{2k} z^{-2}}$$

this product represents a general distribution of poles and zeros and suggests a set of structures consisting of a cascade of first and second-order subsystems. There is considerable freedom in the design of the subsystems, but it is best to realize the systems using a minimum of storage.

The expression of $H(z)$ in this form indicates the presence of poles and zeros in pairs. If poles and zeros

are not present in pairs, one of the coefficients B_{2k} or a_{2k} will be zero as appropriate. An implementation of such a cascade structure with the use of minimum memory can be obtained through a direct form II realization of each second order subsystem using techniques similar to the direct form implementation utilized previously. A cascade realization of a sixth-order system, such as the ASQ-81 system, using a direct form II realization of each second order subsystem would appear as in Figure 4.2 below.

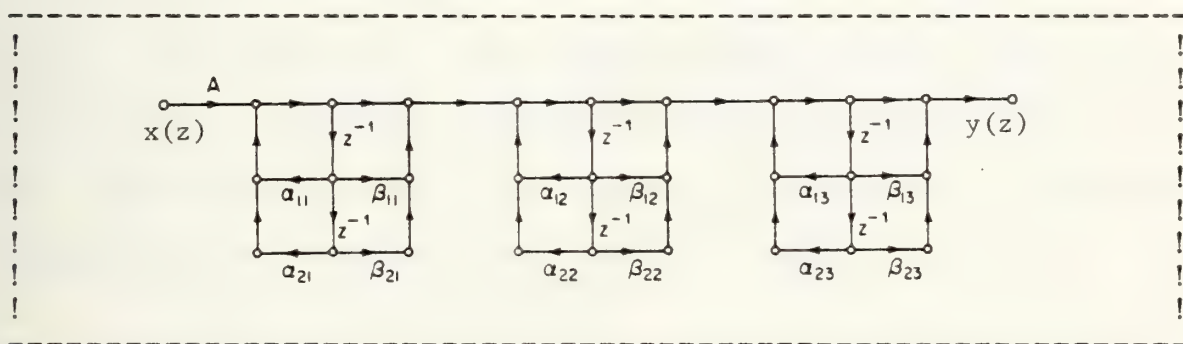


Figure 4.2: Cascade Structure With a Direct Form II Realization of Each Second Order Subsystem. [Ref. 7]

There is, theoretically, considerable flexibility in the manner in which the poles and zeros are paired together and in the order in which the resulting second-order subsystems are cascaded. However, although all such pairings and orderings are equivalent for infinite-precision arithmetic, they may differ considerably in practice owing to finite word length effects of roundoff and truncation.

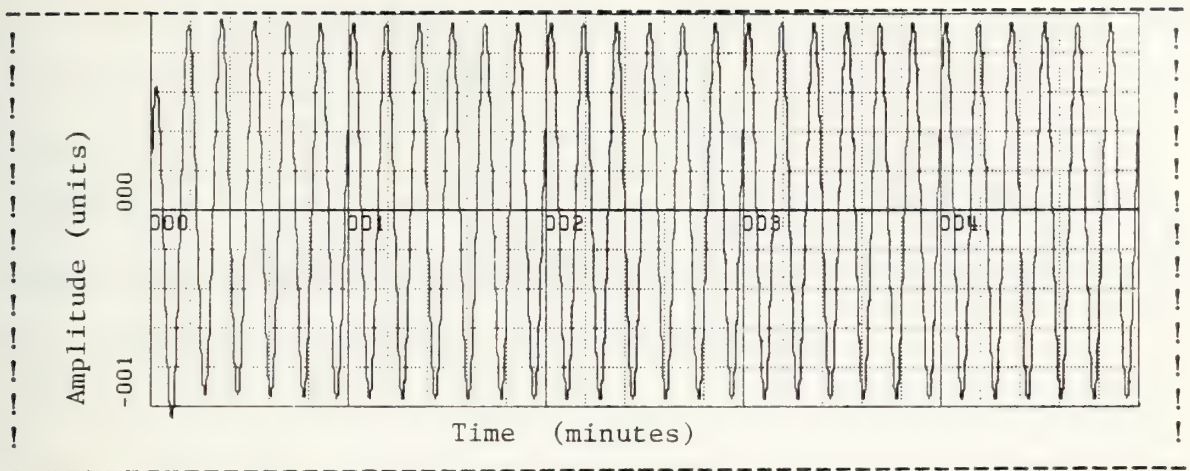


Figure 5.1: Output of First Stage Filter of Digital Filter Computer Program With Sinusoidal Input in Simulation.

Unfortunately, the second stage output of the filter showed an instability within the program design, indicated by the output of the filter being a sinusoid of increasing magnitude, as indicated in Figure 5.2 below.

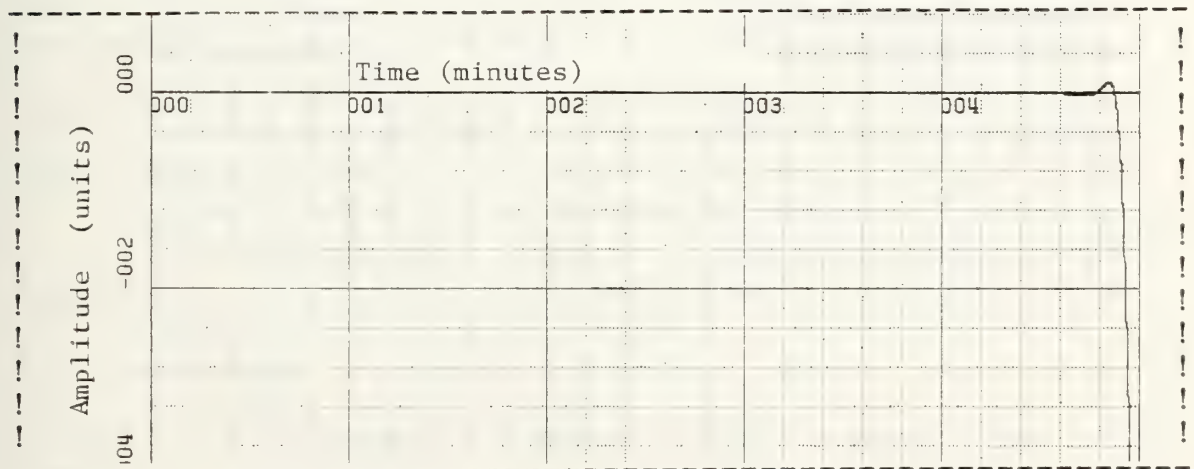


Figure 5.2: Output of Second Stage Filter Design With Input of a Sinusoid.

The stability of the third stage of the filter design was investigated by inputting the sinusoid directly to the third filter, and found to be stable. A check of the derivation of the equations, coefficients, and programming steps of the second (unstable) filter of the design failed to indicate the cause of the instability.

Computation of the poles of the z transfer function, $H(z)$, of the second stage of the filter confirmed the instability of the design. The poles were computed to be: $0.92 \pm 0.1218i$, $1.07 \pm 0.1340i$, 0.8611 , and 1.1564 . Of these six poles, three lie outside the region of convergence for the z -plane, that is, within the unit circle discussed previously in Chapter IV.

The second stage of the filter was therefore redesigned using the cascade form of the direct form realization (direct form II), and tested in simulation. A copy of the software used in the simulation is enclosed in Appendix F.

The output of all three filter stages of the program were stable, as indicated in Figures 5.3 through 5.7 below. The amplitude decrease and phase shift expected were observed. The "damped overshoot" of the second stage output is due to the fact that, for values of the input function prior to time zero in the simulation, utilized in the input-output signal difference equations for the filter, the input signal was set at 0. This resulted in an instantaneous

change of the input signal from 0 to the finite value introduced in the simulation at time 0+. The "overshoot" of the filter is the filter's attempt to "match" this instantaneous jump in magnitude of the input signal. When the input signal to the filter in the simulation is zero at time zero, this overshoot effect does not occur.

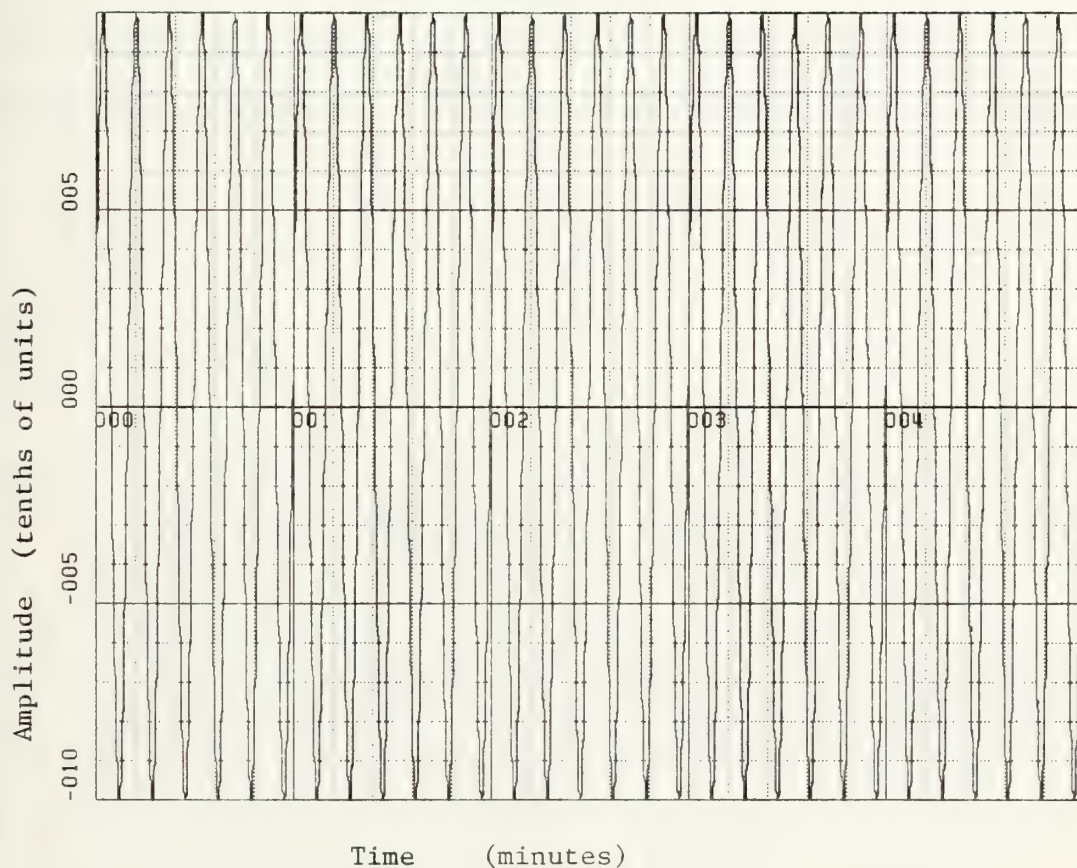


Figure 5.3: Input Signal to Digital Filter Program. A Sinusoid of Frequency 0.1 HZ and Amplitude ± 1 .

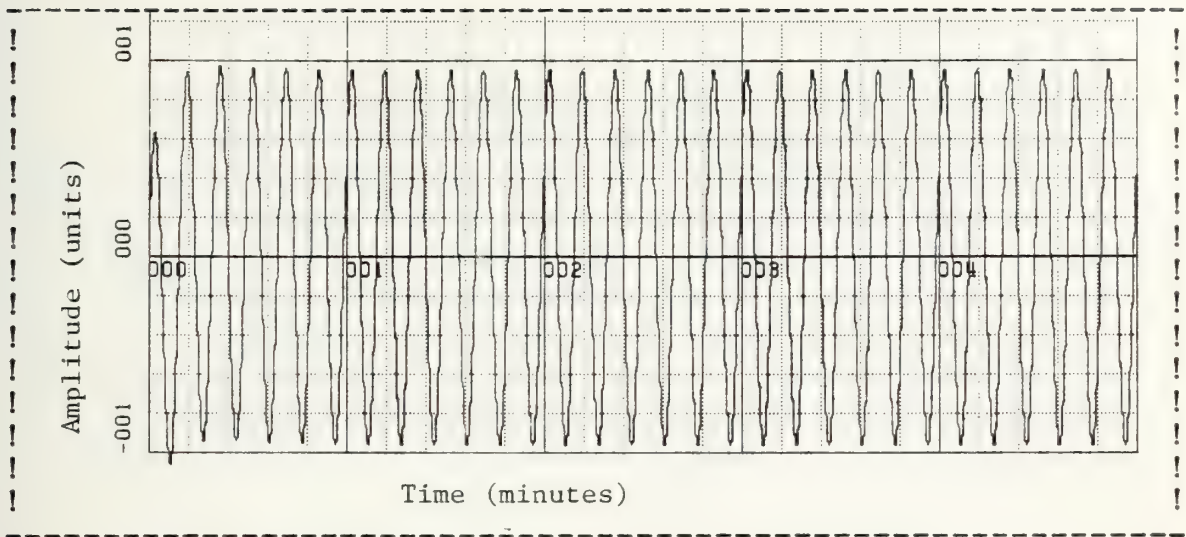


Figure 5.4: Output of First Stage of Digital Filter Program.

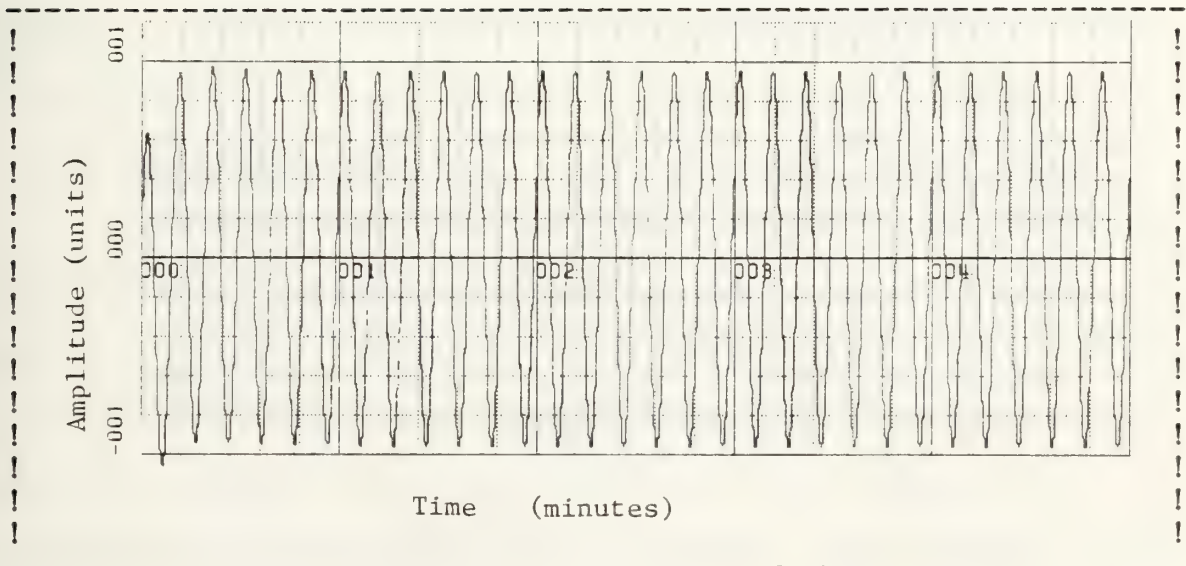


Figure 5.5: Output of Second Stage of Digital Filter Program.

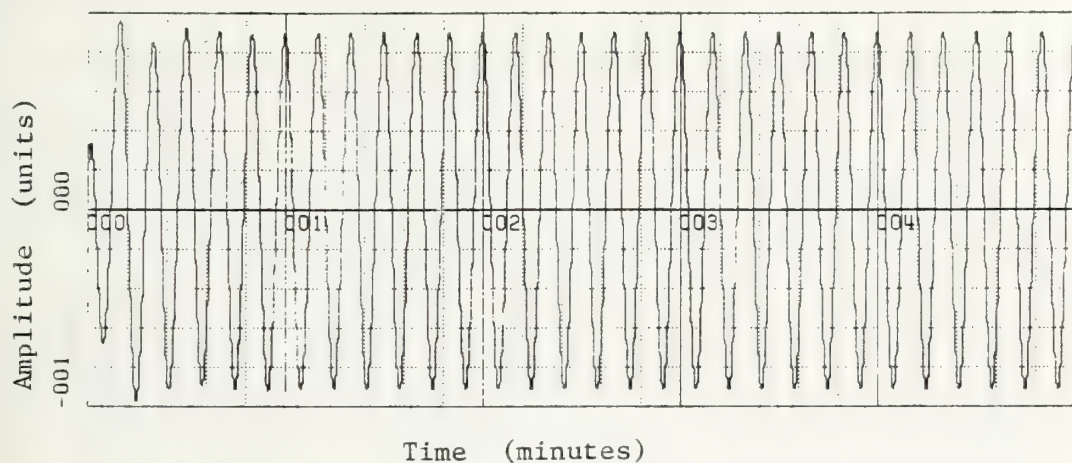


Figure 5.6: Output of Third Stage of Digital Filter Program.

This simulation was run with inputs of sinusoids of various frequencies in order to check the stability of the filter design at frequencies throughout the operating range of the AN/ASQ-81 magnetometer. In all cases, the design was stable, and the expected amplitude changes and phase shifts occurred.

2. Noiselike Inputs

The simulation was also run with inputs consisting of a sinusoid of a frequency which should be passed through the AN/ASQ-81 added to sinusoids of frequencies which should have been filtered by the magnetometer and random noise. The filter performed as expected, with the sinusoid of a passable frequency passed by the filter, and spurious noise and sinusoids attenuated severely. The results of a simulation consisting of a sinusoid of passable frequency, a

filterable sinusoid, and uniformly distributed random noise, all of amplitude ± 1 , are presented in Figures 5.7 through 5.10 below.

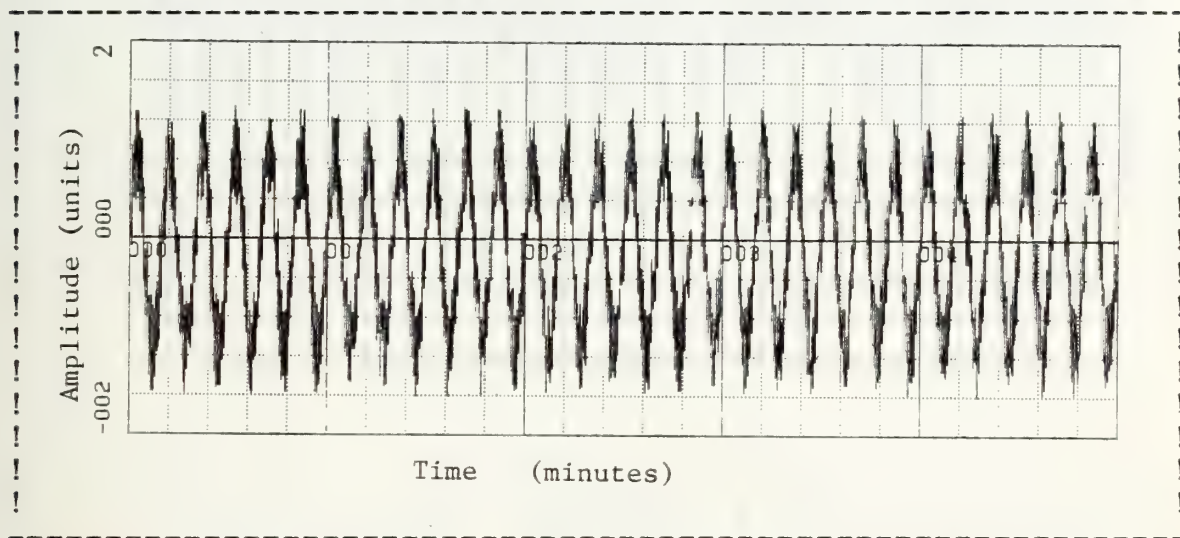


Figure 5.7: Input to Filter - 0.1 HZ Sinusoid, 10 HZ Sinusoid, Uniformly Distributed Random Noise of Amplitude ± 1 .

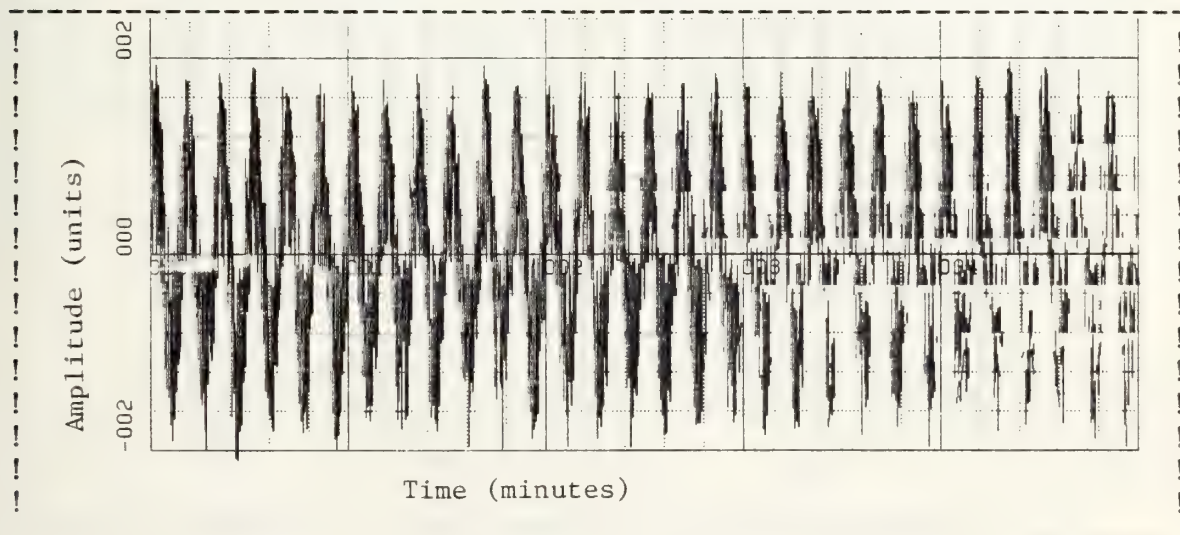


Figure 5.8: Output of First Filter Stage.

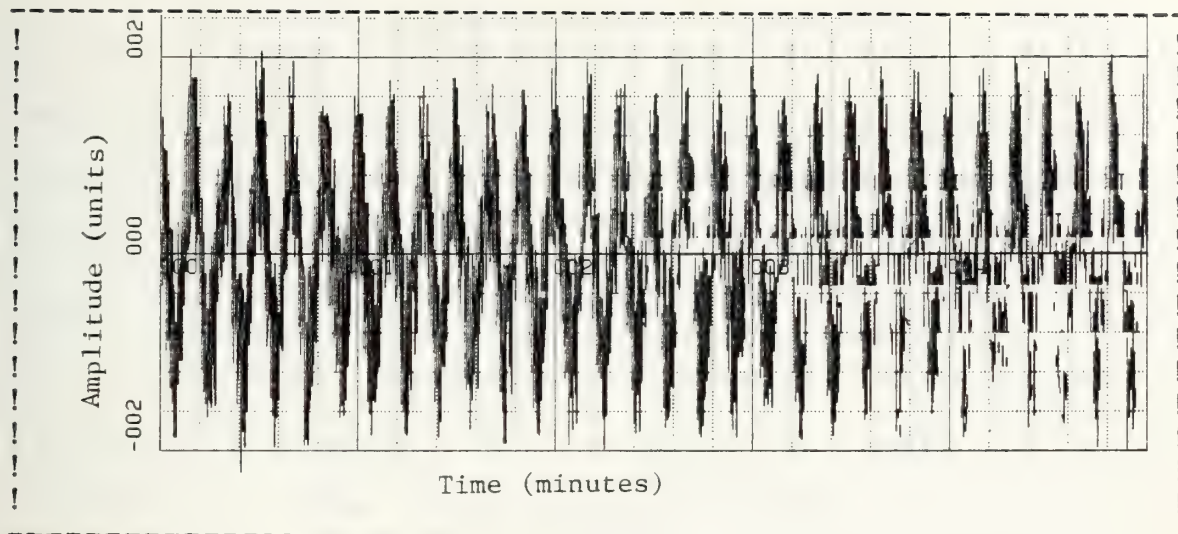


Figure 5.9: Output of Second Filter Stage.

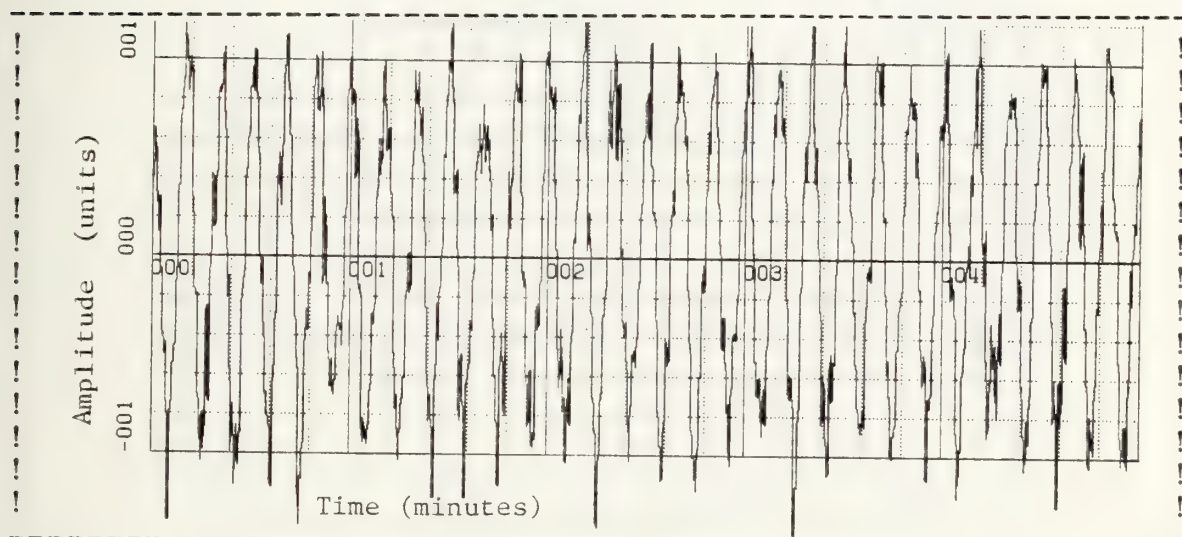


Figure 5.10: Final Filter Stage Output.

As can be seen, the digital filter program succeeds in filtering out random noise and signals of frequency components above the band pass of the magnetometer.

In order to ensure that the digital filter representation of the magnetometer has the same amplitude

versus frequency characteristics of the AN/ASQ-81 magnetometer, a simulation program was written which inputs sinusoids of varying frequencies and computes the Root Mean Square (RMS) value of the filter output and the signal input, then computes the decibel (dB) attenuation of the filter at that frequency. A copy of this program is included in Appendix G. A plot was made of the dB attenuation versus frequency for the filter and compared

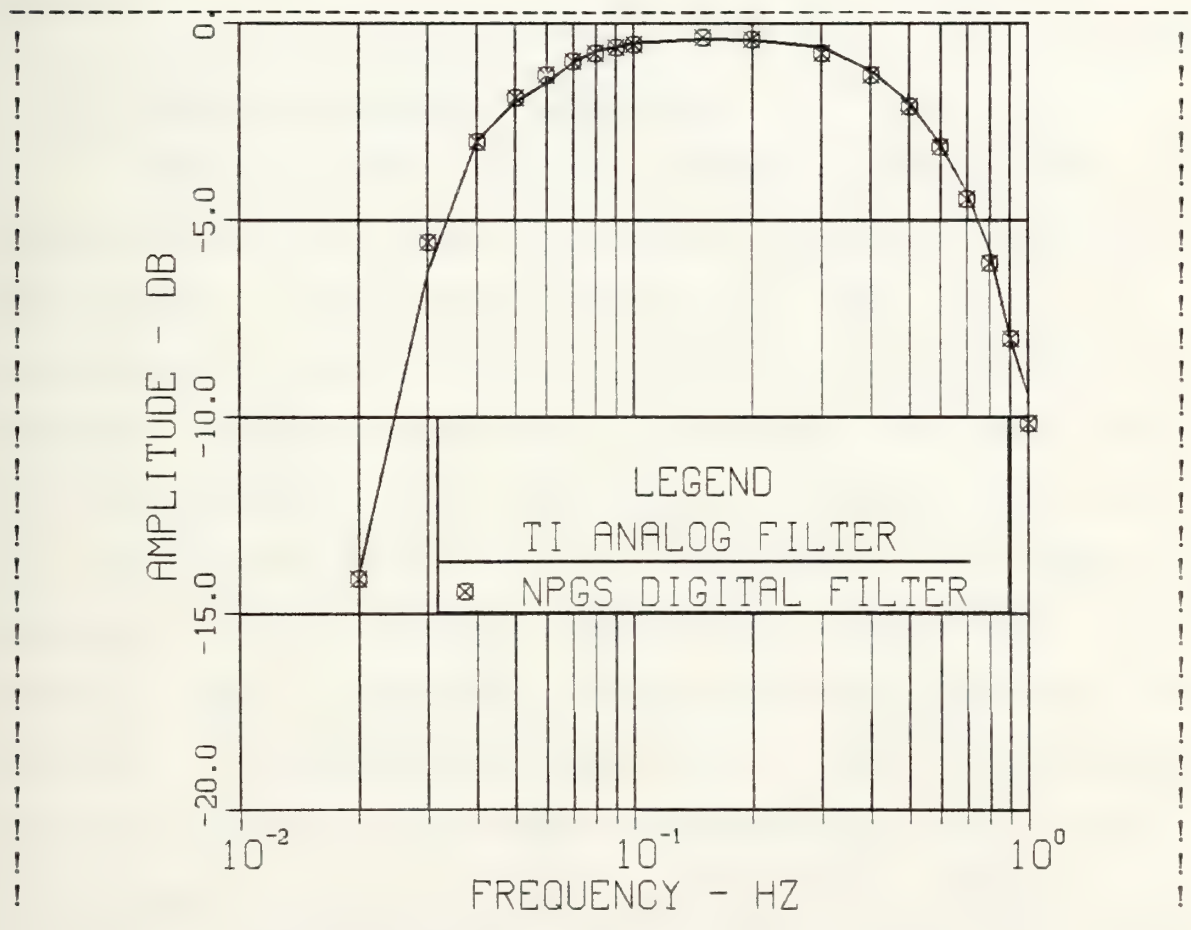


Figure 5.11: Plot of Attenuation Versus Frequency for Sinusoidal Inputs for Digital Filter and Analog Filter.

with the measured frequency performance of the AN/ASQ-81 magnetometer, which was supplied by Texas Instruments, Inc., and does not include the effects of the fixed high pass filter. Consequently, the data shown in Figure 5.11 is a comparison of the data supplied by Texas Instruments and the output of the test program, which also does not include the fixed high pass filter. As can be seen, the performance of the filter is extremely similar to that of the magnetometer itself.

3. Anderson Function Simulations

The next step in the simulation phase was the introduction to the filter of Anderson function simulations. The shape of the signal amplitude of the output of a magnetometer passing through the sphere of influence of a magnetic anomaly (submarine) is a function of the dip angle of the geomagnetic field, the magnetic heading of the track of the magnetometer (or the aircraft), the magnetic heading of the anomaly (submarine) dipole, and the lateral range between the magnetometer (aircraft) and the anomaly. Anderson functions [Ref. 9] are mathematical representations of three basic components of signals which, when taken in various linear combinations, describe the shape of these anomaly signals. The equations for the Anderson functions are:

$$\begin{aligned}
 \text{(First Anderson Function)} \quad f_0 &= 1/(1 + B^{2.5/2}) \\
 \text{where} \quad B &= \frac{(\text{velocity}) \times (\text{time})}{\text{range at CPA}},
 \end{aligned}$$

or, a dimensionless parameter defined as the distance traveled along the magnetometer (aircraft) track divided by the slant range at closestpoint of approach (CPA)

$$\begin{aligned}
 \text{(Second Anderson Function)} \quad f_1 &= B \times f_0^2 \\
 \text{(Third Anderson Function)} \quad f_2 &= B \times f_1 = B^3 \times f_0
 \end{aligned}$$

The Anderson functions were introduced into the filter program in a noise-free signal environment in order to observe the output signal and ensure that it was a "MAD-like" signal. A rigorous determination of the actual output signal would have been extremely difficult to obtain, so a comparison was made with the output of a computer simulation program provided to NPS by Mr. Joe Rice of Texas Instruments. When the sampling rate of the program was adjusted to equal that of the Texas Instruments program, 8 HZ, the two program outputs were observed to be very similar. The Anderson function simulation inputs and outputs of the program are depicted in Figures 5.12 through 5.18. The Texas Instruments program outputs were obtained in the form of time series plots of discontinuous data points, and were therefore not conducive to replotting for comparison.

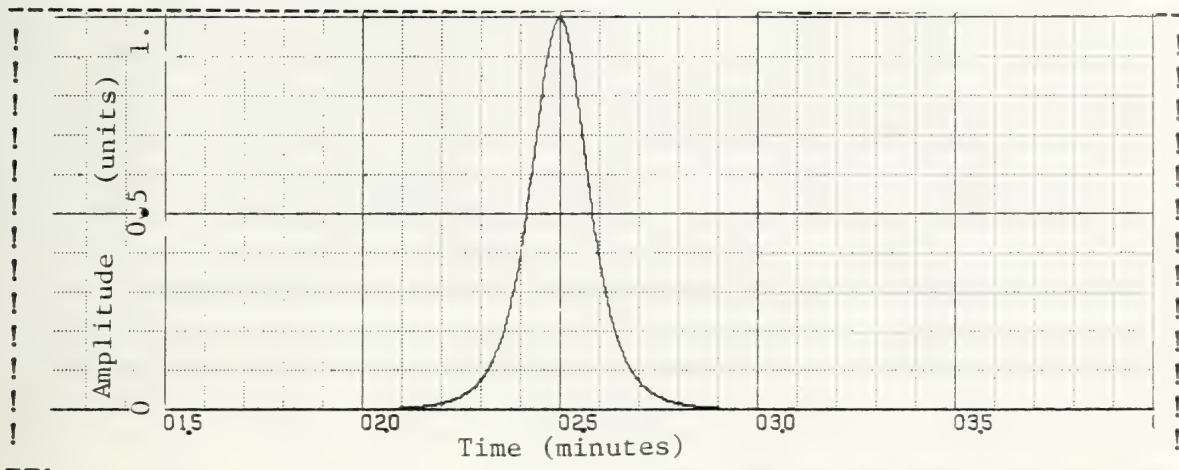


Figure 5.12: First Anderson Function Input. CPA at Time 2.5 Minutes.

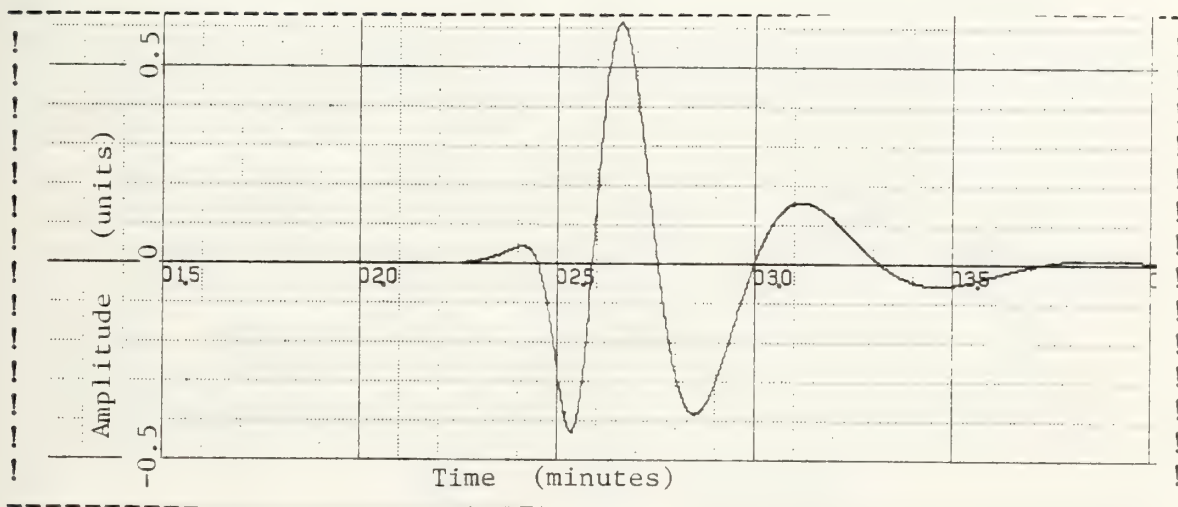


Figure 5.13: Filter Output for First Anderson Function Input.

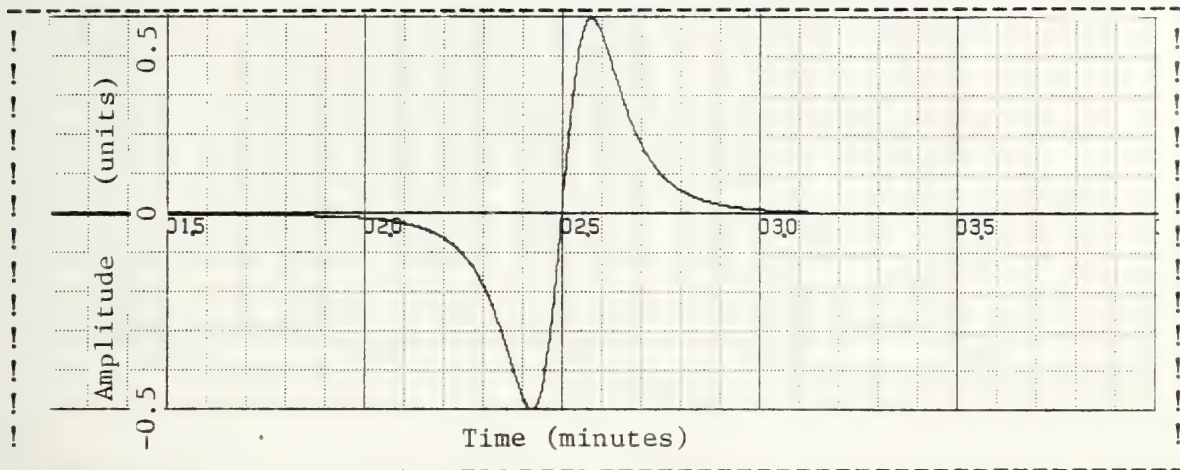


Figure 5.14: Second Anderson Function Input. CPA at Time 2.5 Minutes.

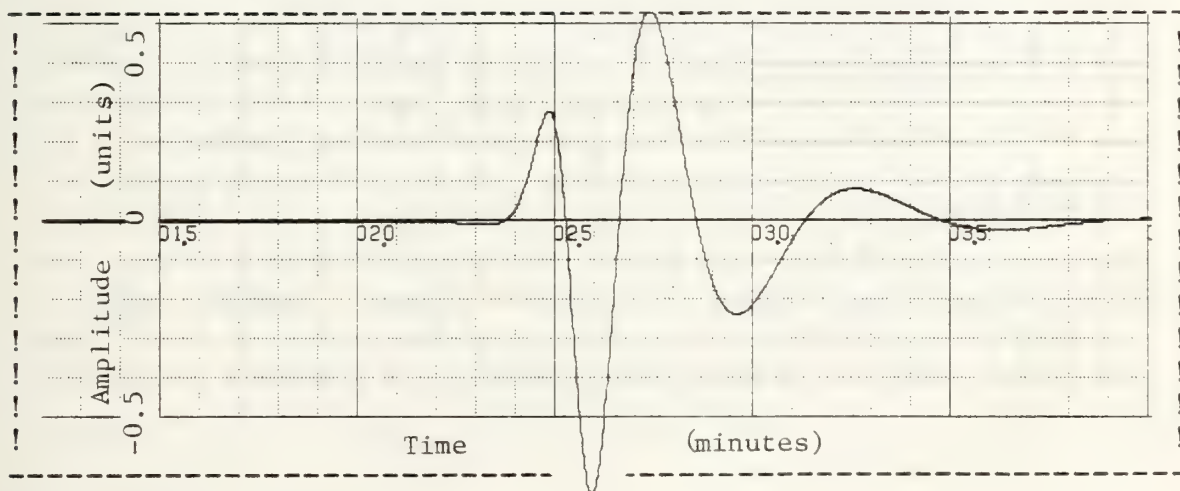


Figure 5.15: Filter Output for Second Anderson Function.

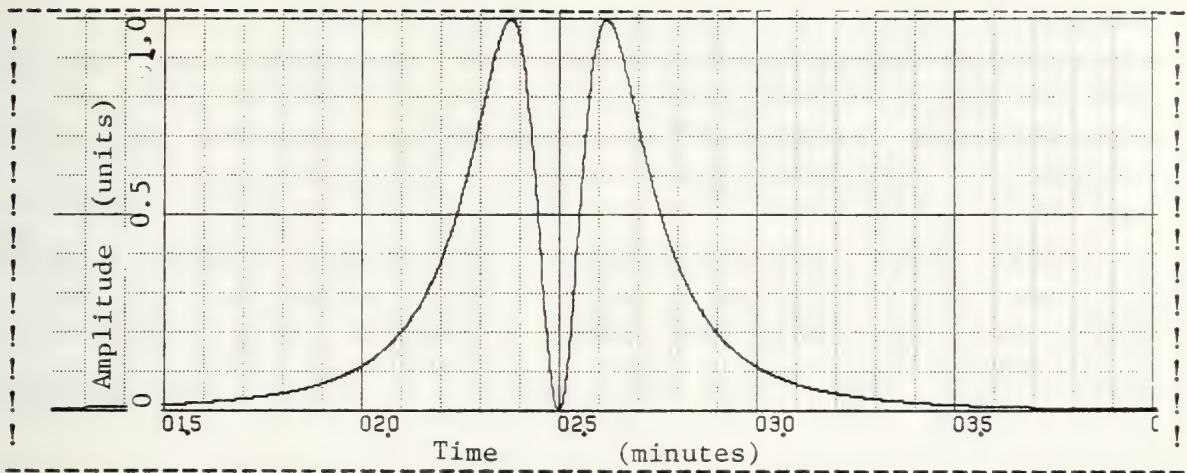


Figure 5.16: Third Anderson Function Input. CPA at Time 2.5 Minutes.

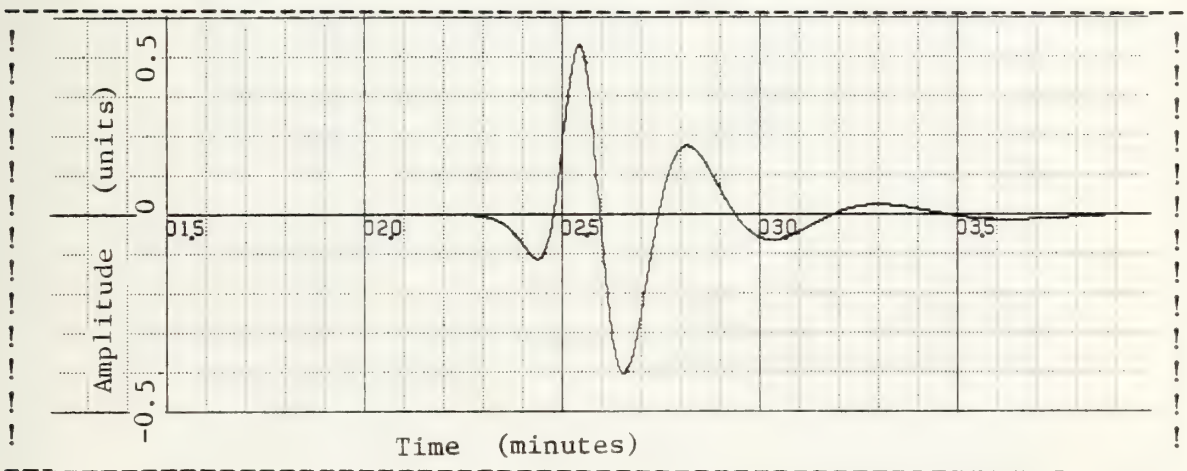


Figure 5.17: Filter Output for Third Anderson Function.

The filter output for all three Anderson function inputs did appear to be "MAD-like" signals, and did closely resemble the simulation output obtained from Texas Instruments, Inc.

4. Impulse Function Response

The response of the filter program was also observed when the input was a unit impulse function. Again, the

output was compared to that of the Texas Instruments' computer program. The outputs of the two programs were observed to be, again, very similar, as can be seen in Figure 5.18, where the response of the NPGS filter is represented by a solid line and that of the Texas Instruments filter by a chain-dash line. The abrupt "jumps"

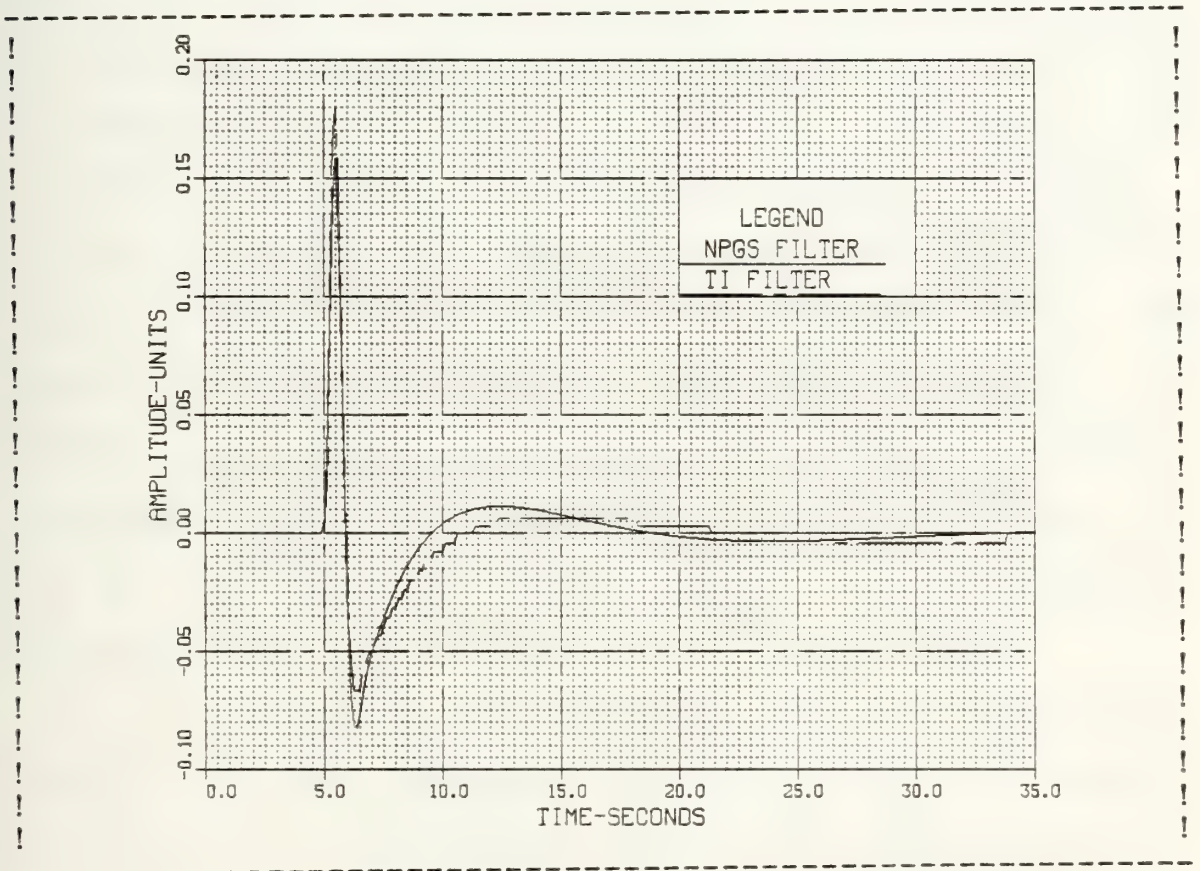


Figure 5.18: Impulse Response of Filters.

of the Texas Instruments response are due to the translation of the output plot supplied to this plot. The plot supplied by Texas Instruments was, again, discontinuous points of poor resolution, and it was necessary to interpolate values

in order to generate Figure 5.18. This resulted in the broken appearance of the plot. Even so, the similarity of the outputs can be observed.

B. EQUIPMENT SETUP

Following the simulation phase of the experiment, actual magnetic field measurements were introduced to the filter in order to test the response of the filter. Magnetic field measurements were made at the La Mesa field test site near the Naval Postgraduate School in Monterey. The output of an AN/ASQ-81 magnetometer, a Schonstedt magnetic field sensor, and the school's coil sensor, oriented along the Earth's magnetic field, were pulse code modulated (PCM) and transmitted via VHF radio to recording devices at the Postgraduate school. The recording of a two hour long data collection period was transferred to digital data tape for use by the school's IBM3033 general purpose mainframe computer.

In the first test of the digital filter program, the output of the Schonstedt sensor, which represents fluctuations of the Earth's total field, was used as the input to the computer program. A comparison of the output of the computer program, with this approximation to the total field fluctuations as input, to the output of the AN/ASQ-81 should provide an indication of the proper functioning of the computer filter program. The results of

the test are shown in Figures 5.19 through 5.21 on the pages following. Figure 5.19, the Schonstedt sensor output, shows several instances of PCM dropouts, that is, occasions where the pulse code modulation signal was not correctly read by the computer for some reason. At such occurrences, the data point value used by the computer is a random number and does not reflect the true value of the data. The problem with these PCM dropouts is that the computer does not recognize them as invalid data points and will use them in computations. This can (and does) cause problems in the computation of Fourier transforms, spectral characteristics, etc. Additionally, this will also impact the proper functioning of the digital filter program which is the subject of this thesis. PCM dropouts are visible at times 6, 8, 142, and 220 through 226 seconds on the plot of the Schonstedt sensor output. An examination of Figure 5.20, the filter program output, reveals the programs attempt to "follow" these PCM dropouts. It should be recalled that previous simulations indicated the filter's tendency to "follow" sudden changes in the input signal, with a relaxation time required for the filter to steady out. This effect is apparent in the output of the filter program at times corresponding to those of the PCM dropouts in the Schonstedt sensor's time series plot. It can be seen that this overshoot tendency resulted in an output significantly different from the actual AN/ASQ-81 output at these times.

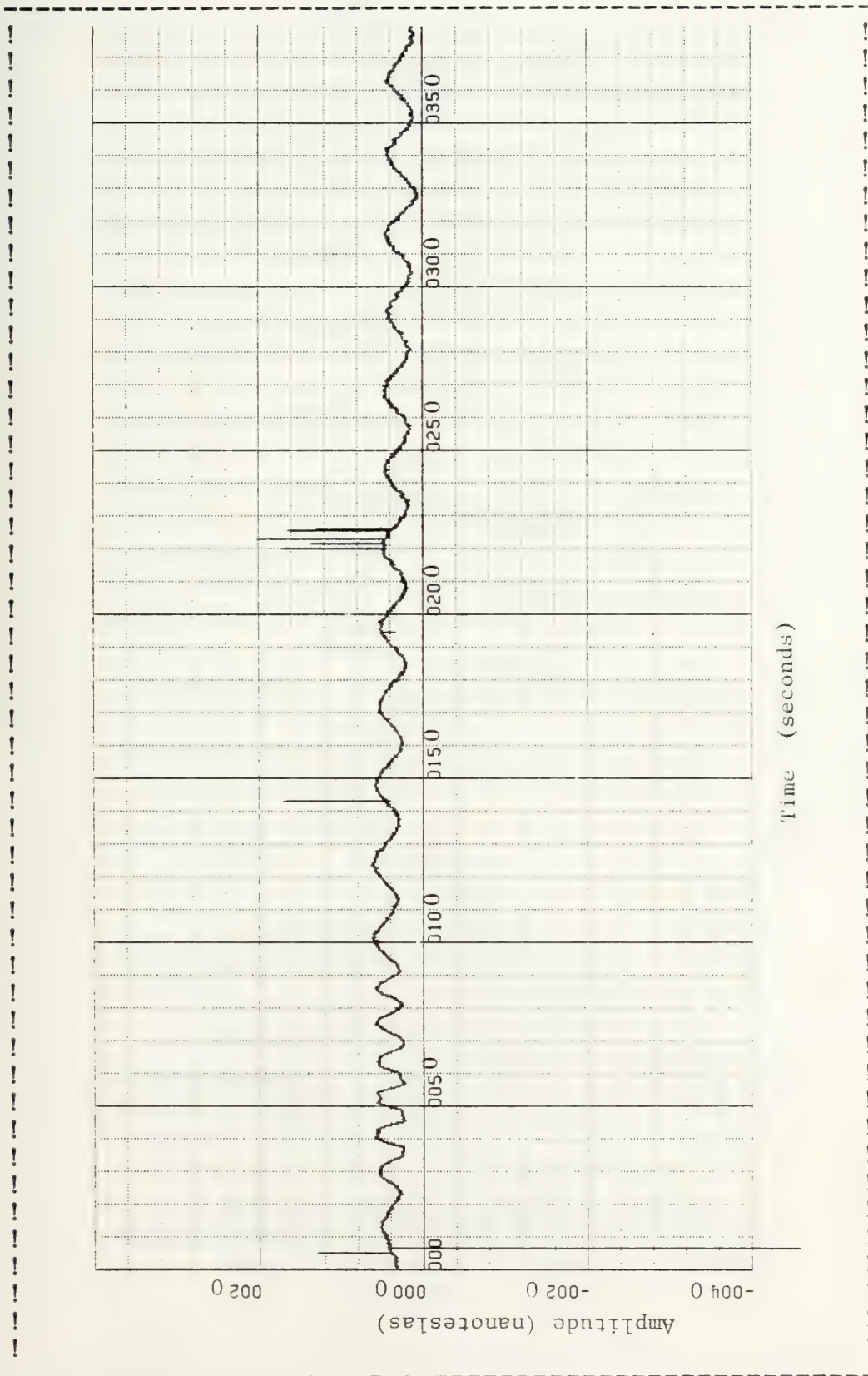


Figure 5.19: Schonstedt Coil Time Series Output

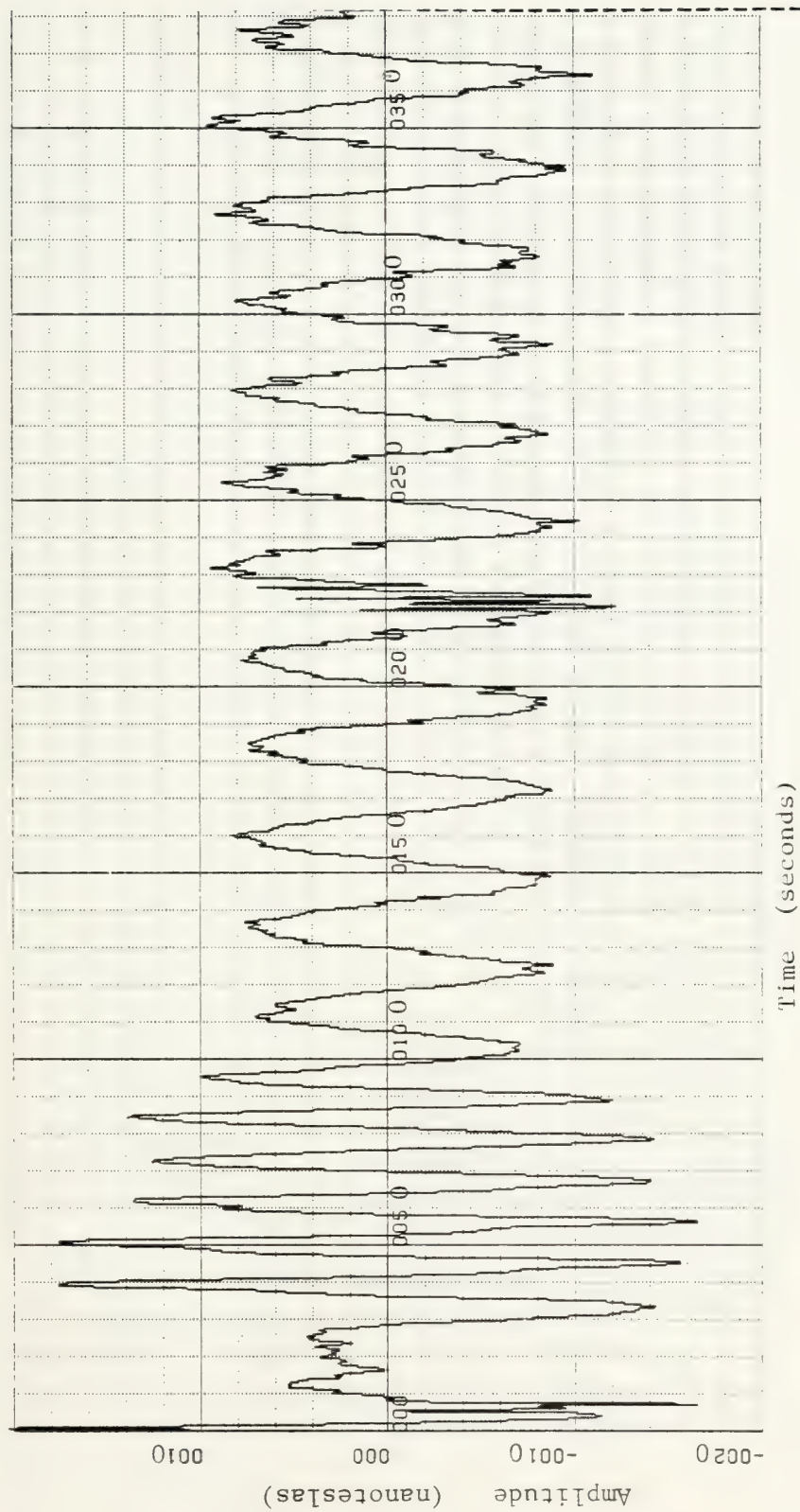


Figure 5.20: Program Time Series Output With Schonstedt Coil as Input

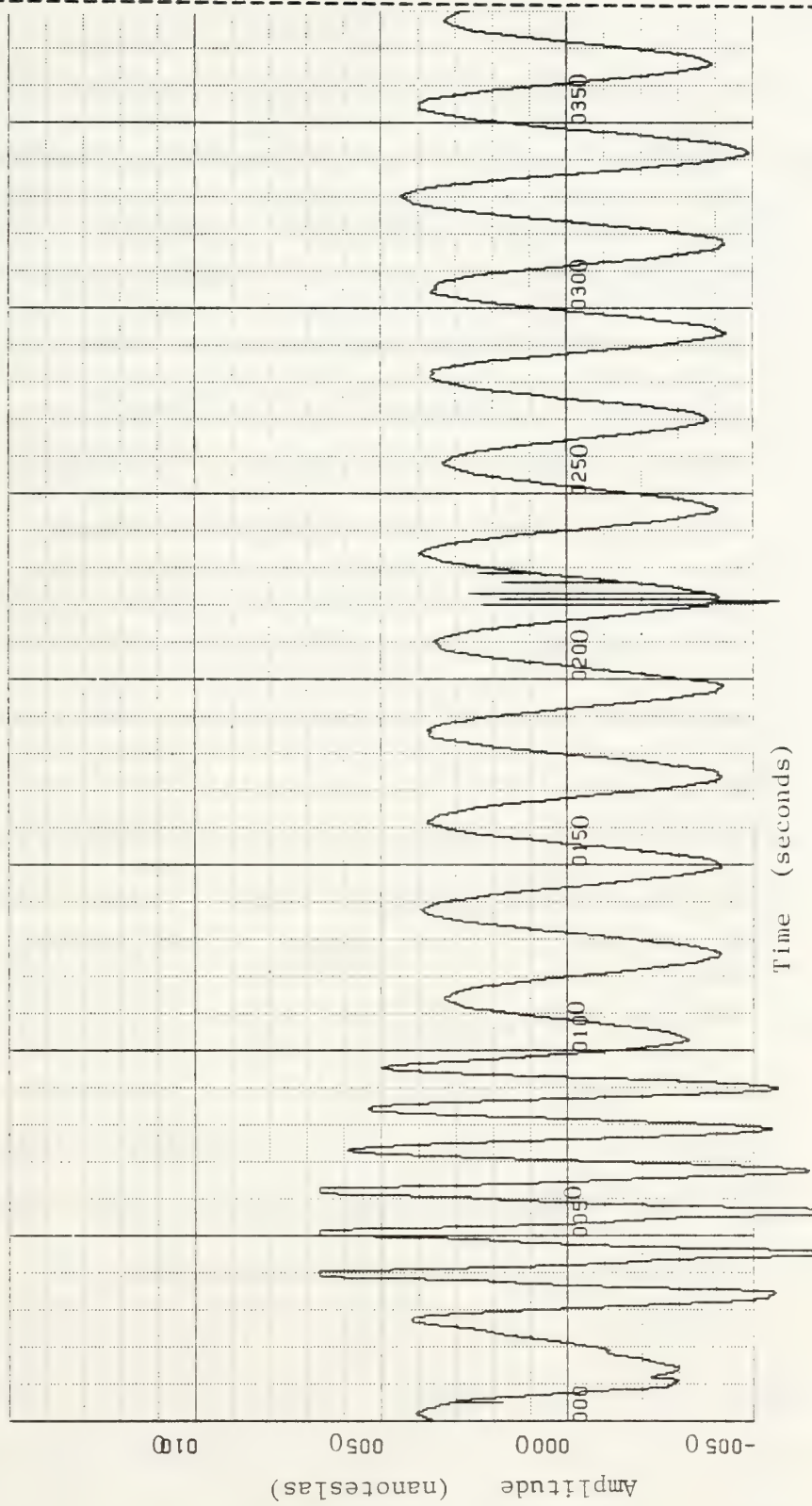


Figure 5.21: ASQ-81 Time Series Output

If the PCM dropout induced differences are neglected, it can be seen that the shape of the output of the filter program is remarkably similar to that of the AN/ASQ-81 magnetometer, although noisier. Note that the output of the AN/ASQ-81 magnetometer exceeded the maximum voltage amplitude which the pre-amplifiers of the data collection system were able to handle and resulted in a truncated signal from time 40 to time 60 seconds. It can still be seen, however, that the filter program output is very similar to the time signal which would have been displayed without this truncation.

It should be noted that the amplitudes of the time series signals of the program output and the AN/ASQ-81 magnetometer differ considerably. In the case of the program output, the peak amplitudes are on the order of 1.4 nanoteslas along the vertical scale, while the peak amplitudes of the output of the AN/ASQ-81 magnetometer are on the order of 0.7 nanoteslas along the vertical scale. This is because the input signal to the filter program is an approximation to the total field difference time series signal, and some amplitude difference could reasonably be expected. The intent of this initial test was to investigate the output time series shape, and an exact correlation was not expected. It is worth noting that the digital filter program will perform its function on any time series signal, regardless of units. This means that a signal may be

operated upon either before or after conversion from whatever units it was originally measured to magnetic field strength units.

Therefore it appears that the digital filter program is functioning properly. When a close approximation to the fluctuations of the total field time series signal is used as the input to the computer program, the output of the program is similar to the time series output of an AN/ASQ-81 magnetometer.

The final stage in the testing process was a conversion of the time series output voltage signal of the coil antenna sensor, which was aligned along the Earth's magnetic field, into a total field fluctuation time series representation for the same time period as before, and then to use this as the input to the digital filter program. A comparison of the resultant time series output of the program with the actual AN/ASQ-81 magnetometer output would validate the proper functioning of the program.

Conversion of the time series antenna sensor output voltage signal into total field fluctuations in nanoteslas was accomplished through the use of a computer program designed by Capt. Kurt Stevens, USAF, a student at the Naval Postgraduate School, as his Master's thesis [Ref 10]. The output voltage time series is stored in an array, then a Fourier transform is performed on the stored data, resulting in the Fourier spectrum of the data. This spectrum is

corrected for the characteristics of the coil antenna sensor to obtain the Fourier spectrum of the total field data. A reverse Fourier transform gives the time series signal for total magnetic field in nanoteslas.

This time series signal was used as the input to the digital filter program and compared with the output of the AN/ASQ-81 magnetometer. Figures 5.22 through 5.24 show the raw coil antenna data, the total field time series data, and the program output time series for a 6 minute period of the test. Figure 5.22 shows the raw coil antenna data series. The number of PCM dropouts should be noted, as these will influence the performance of the filter program. Figure 5.23 shows the computed total field time series. Note that the PCM dropouts evident on the raw time series plot are evident on the computed total field time series plot also, and thus inputted to the filter program as valid data points. Additionally, there are two "jumps" in the plot of total field fluctuation (Figure 5.23) which are also inputted to the filter program as valid data points. These "jumps" are located at 128 and 256 seconds and are caused by the method of processing blocks of data for the conversion to total field fluctuation. A block of 128 seconds of data is processed at a time, and the results of each block are stored in an array. This results in a slight amplitude difference between the last data point of one block and the

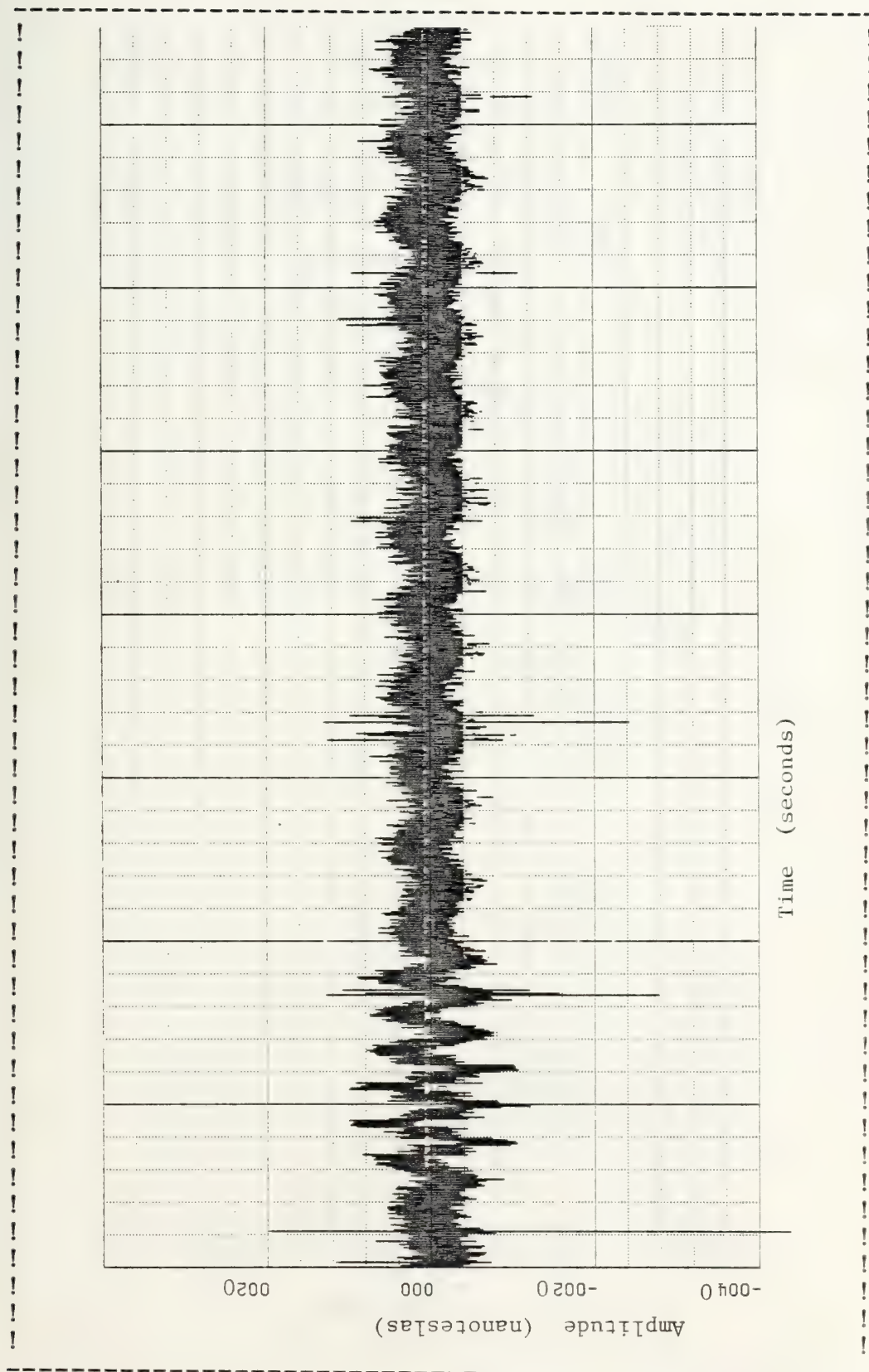


Figure 5.22: Raw Coil Antenna Time Series Output

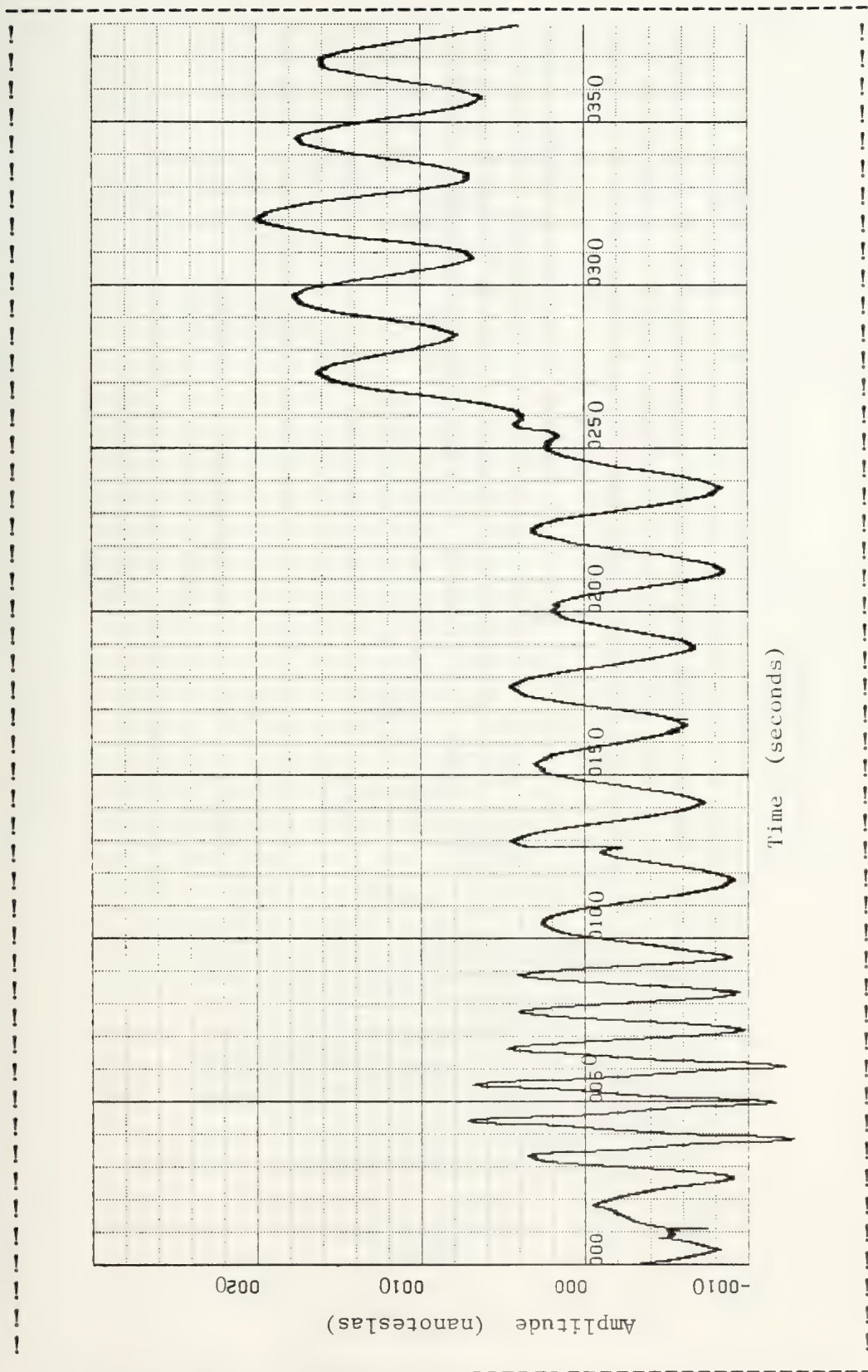


Figure 5.23: Coil Antenna Difference Field Time Series Output

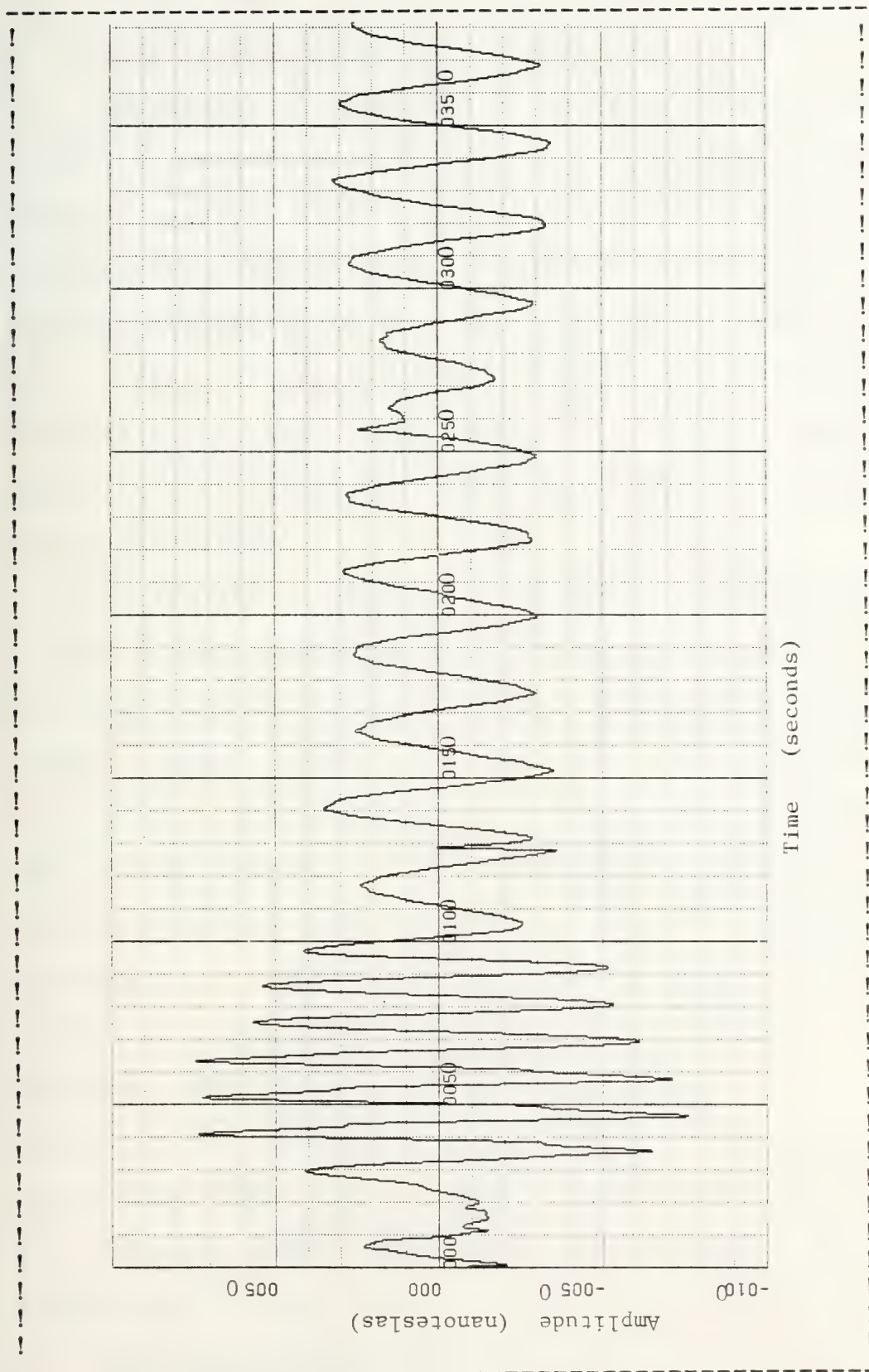


Figure 5.24: Program Time Series Output With Coil Antenna Difference Field Time Series as Input.

first data point of the next block of data. This slight difference is manifested as a signal jump.

A comparison of Figures 5.24 (program output) and 5.21 (AN/ASQ-81 output) show that the filter program gives a time series output very similar to that of the actual magnetometer. The first 20 seconds of the program output is somewhat dissimilar to that of the AN/ASQ-81, due either to the initial "start up" delay of the filter program or to distortion of the total field fluctuation time series. There is a PCM dropout at time 11 seconds which contributed to the distortion.

Following this, however, it can be seen that the program output is very similar to that of the magnetometer, except at 128 and 256 seconds, which show the effects of the false signal jumps caused by the total field fluctuation conversion. There is also a noticeable four to five second time delay between the AN/ASQ-81 output and that of the filter program. As this delay is not evident in a comparison of the AN/ASQ-81 output and that of the filter program with the Schonstedt sensor as the input, it can be inferred that this time delay is caused either by the program which converts the raw coil data to total field fluctuation data, or by a phase (and hence time) change of the voltage signal due to the coil sensor itself. A comparison of the raw coil data in Figure 5.22 to the converted coil data in Figure 5.23 indicates no time shift,

and hence the deduction can be made that there is a time delay inherent within the coil sensor itself.

Other than the differences of the four to five second time delay and the distortions caused by the false signal jumps, the program output is extremely similar to the output of the AN/ASQ-81 magnetometer.

VI. CONCLUSIONS

The intent of this thesis was to design and test a digital filter computer program which would, when given a time series input of fluctuations in the total magnetic field, deliver an output time series representation of the output of an AN/ASQ-81 magnetometer. This purpose has been realized.

The computer program contained in Appendix I has been proven to output a time series signal which is very similar to that of the magnetometer. The major limitations of the output signal are a finite time delay of about five seconds between the AN/ASQ-81 magnetometer signal and the output signal of the program, a sensitivity of the program to false data points such as those caused by PCM dropouts and false signal jumps caused by processing large data blocks, and the inherent limitations of the program caused by its dependence on the use of digital data tapes and the IBM 3033 mainframe computer.

The five second time delay is not considered to be an important limitation to the program, as it was intended as a research tool for programs currently in progress at the Naval Postgraduate School. Instances where this time delay might become important would be in areas of simultaneous comparison of the program output signal with an actual magnetometer, in target location algorithms using time

delays, or in correlation studies between different sensors. In correlation studies using coil sensors, the effects of the time delay would cancel out, as all coil outputs would be similarly delayed. In target location algorithms, the target location errors due to the time delay could be adjusted for simply while in computer simulation, and flight testing could not reasonably be accomplished without the use of an actual magnetometer as the sensor. Lastly, in a comparison of the program output with an actual sensor, the time delay can, again, be compensated for. In short, these limitations are not considered excessive, especially as the apparent cause for the delay is not the filter program.

In the primarily intended purpose of the filter program, magnetic noise studies, the time delay is not considered to be a problem.

The problem of false data points caused by PCM dropouts and signal jumps due to conversion to total field fluctuations is more serious. False data points cause inaccuracies in the output time series and could adversely affect later projects. Unfortunately the PCM dropout problem is one which is endemic to the data collection system presently being used at the postgraduate school, and not to the filter program itself. It is imperative that users of this program are aware of the PCM dropout problem and of the effects it may entail upon their specific

research. A large number of PCM dropouts in a time series could render that series unusable. Similarly the case of the false data jumps caused by conversion to total field fluctuations is not within the filter program. Further investigation of this problem is necessary in order to eliminate it.

The last problem, that of reliance upon the digital data tape/IBM 3033 computer system, is, like the PCM problem, one which is not endemic to the filter program but rather to the data collection system being used. A change of data collection system may, at some future time, remove the reliance upon the PCM/digital tape/IBM 3033 data system (and hence too the data block conversion problem which results in false data jumps), but this is unlikely at this time. Users should be aware of this dependence and of possible effects upon specific research projects.

APPENDIX A

AN/ASQ-81 FILTER TRANSFER FUNCTIONS

Fixed High Pass Transfer Function:

$$H(S) = \frac{80 S^2}{80 S^2 + 20 S + 1}$$

Selectable High Pass Transfer Functions:

A. 0.04 HZ $H(S) =$

$$\frac{40.82834 S^2}{40.82834 S^2 + 12.52096 S + 1} \times \frac{45.28317 S^2}{45.28317 S^2 + 11.00999 S + 1}$$

$$\times \frac{57.576688 S^2}{57.576688 S^2 + 7.41498 S + 1}$$

B. 0.06 HZ $H(S) =$

$$\frac{18.14591 S^2}{18.14591 S^2 + 8.34727 S + 1} \times \frac{20.12587 S^2}{20.12587 S^2 + 7.33999 S + 1}$$

$$\times \frac{25.58964 S^2}{25.58964 S^2 + 4.94332 S + 1}$$

C. 0.08 HZ $H(S) =$

$$\frac{10.20708 S^2}{10.20708 S^2 + 6.26045 S + 1} \times \frac{11.32080 S^2}{11.32080 S^2 + 5.50500 S + 1}$$

$$\times \frac{14.39417 S^2}{14.39417 S^2 + 3.70749 S + 1}$$

$$\begin{aligned}
 \text{D. } & \underline{0.10 \text{ HZ}} \quad H(S) = \\
 & \frac{6.53253 S^2}{6.53253 S^2 + 5.00836 S + 1} \times \frac{7.24531 S^2}{7.24531 S^2 + 4.40400 S + 1} \\
 & \times \frac{9.21227 S^2}{9.21227 S^2 + 2.96599 S + 1}
 \end{aligned}$$

Selectable Low Pass Transfer Functions

$$\begin{aligned}
 \text{A. } & \underline{0.2 \text{ HZ}} \quad H(S) = \\
 & \frac{1}{0.3143 S^2 + 1.0741 S + 1} \times \frac{1}{0.2501 S^2 + 0.6209 S + 1}
 \end{aligned}$$

$$\begin{aligned}
 \text{B. } & \underline{0.4 \text{ HZ}} \quad H(S) = \\
 & \frac{1}{0.07858 S^2 + 0.53706 S + 1} \times \frac{1}{0.06252 S^2 + 0.31044 S + 1}
 \end{aligned}$$

$$\begin{aligned}
 \text{C. } & \underline{0.6 \text{ HZ}} \quad H(S) = \\
 & \frac{1}{0.03492 S^2 + 0.35804 S + 1} \times \frac{1}{0.02779 S^2 + 0.20696 S + 1}
 \end{aligned}$$

APPENDIX B

AN/ASQ-81 Z TRANSFORM FILTER TRANSFER FUNCTIONS FOR DIRECT FORM I REALIZATION

For fixed high pass filter:

$$H(Z) = \frac{BFHP0 + BFHP1*Z^{-1} + BFHP2*Z^{-2}}{1 - AFHP1*Z^{-1} - AFHP2*Z^{-2}}$$

where BFHP0, BFHP1, BFHP2, AFHP1, AFHP2 are constants tabulated in Appendix D.

For selectable high pass filter:

$$H(Z) = \frac{BSHP0 + BSHP1*Z^{-1} + BSHP2*Z^{-2} + BSHP3*Z^{-3} + BSHP4*Z^{-4} + BSHP5*Z^{-5} + BSHP6*Z^{-6}}{1 - ASHP1*Z^{-1} - ASHP2*Z^{-2} - ASHP3*Z^{-3} - ASHP4*Z^{-4} - ASHP5*Z^{-5} - ASHP6*Z^{-6}}$$

where, for low frequency cutoff of 0.04 HZ:

$$BSHP0 = 0.99471378$$

$$BSHP1 = -5.9682827$$

$$BSHP2 = 14.920707$$

$$BSHP3 = -19.894276$$

$$BSHP4 = 14.920707$$

$$BSHP5 = -5.9682817$$

$$BSHP6 = 0.99471372$$

$$ASHP1 = 5.9894021$$

$$ASHP2 = -14.947051$$

$$ASHP3 = 19.894225$$

$$ASHP4 = -14.894327$$

$$ASHP5 = 5.9472141$$

$$ASHP6 = -0.98945296$$

For low frequency cutoff of 0.06 HZ:

BSHP0 =	0.9920813		
BSHP1 =	-5.9524928	ASHP1 =	5.9841070
BSHP2 =	14.881232	ASHP2 =	-14.920650
BSHP3 =	-19.841643	ASHP3 =	19.841528
BSHP4 =	14.881232	ASHP4 =	-14.841757
BSHP5 =	-5.9524927	ASHP5 =	5.9209919
BSHP6 =	0.99208212	ASHP6 =	-0.98422128

For low frequency cutoff of 0.08 HZ:

BSHP0 =	0.98945806		
BSHP1 =	-5.9367483	ASHP1 =	5.9788144
BSHP2 =	14.841871	ASHP2 =	-14.894276
BSHP3 =	-19.789161	ASHP3 =	19.788959
BSHP4 =	14.841871	ASHP4 =	-14.789365
BSHP5 =	-5.9367476	ASHP5 =	5.8948841
BSHP6 =	0.98945802	ASHP6 =	-0.97901720

For low frequency cutoff of 0.1 HZ:

BSHP0 =	0.98684156		
BSHP1 =	-5.9210493	ASHP1 =	5.9735244
BSHP2 =	14.802623	ASHP2 =	-14.867941
BSHP3 =	-19.736831	ASHP3 =	19.736516
BSHP4 =	14.802623	ASHP4 =	-14.737149
BSHP5 =	-5.9210491	ASHP5 =	5.8688889
BSHP6 =	0.9868415	ASHP6 =	-0.97384065

For selectable low pass filter:

$$H(Z) = \frac{BSLP0 + BSLP1*Z^{-1} + BSLP2*Z^{-2} + BSLP3*Z^{-3} + BSLP4*Z^{-4}}{1 - ASLP1*Z^{-1} - ASLP2*Z^{-2} - ASLP3*Z^{-3} - ASLP4*Z^{-4}}$$

where BSLP0, BSLP1, BSLP2, BSLP3, BSLP4, ASLP1, ASLP2, ASLP3, ASLP4 are constants tabulated in Appendix D.

APPENDIX C

AN/ASQ-81 Z TRANSFORM FILTER TRANSFER FUNCTIONS DIRECT FORM II REALIZATION

For fixed high pass filter:

$$H(Z) = \frac{BFHP0 + BFHP1*Z^{-1} + BFHP2*Z^{-2}}{1 - AFHP1*Z^{-1} - AFHP2*Z^{-2}}$$

where BFHP0, BFHP1, BFHP2, AFHP1, AFHP2 are constants tabulated in Appendix D.

For selectable high pass filter:

$$H(Z) = ASHP1 \times \frac{1 - 2*Z^{-1} + Z^{-2}}{1 - ASHP3*Z^{-1} - ASHP2*Z^{-2}} \times \frac{1 - 2*Z^{-1} + Z^{-2}}{1 - ASHP4*Z^{-1} - ASHP5*Z^{-2}} \times \frac{1 - 2*Z^{-1} + Z^{-2}}{1 - ASHP6*Z^{-1} - ASHP7*Z^{-2}}$$

where ASHP1, ASHP2, ASHP3, ASHP4, ASHP5, ASHP6, ASHP7 are constants and tabulated in Appendix D.

For selectable low pass filter:

$$H(Z) = \frac{BSLP0 + BSLP1*Z^{-1} + BSLP2*Z^{-2} + BSLP3*Z^{-3} + BSLP4*Z^{-4}}{1 - ASLP1*Z^{-1} - ASLP2*Z^{-2} - ASLP3*Z^{-3} - ASLP4*Z^{-4}}$$

where BSLP0, BSLP1, BSLP2, BSLP3, BSLP4, ASLP1, ASLP2, ASLP3, ASLP4 are constants tabulated in Appendix D.

APPENDIX D

Z TRANSFORM REALIZATION DIFFERENCE EQUATIONS

With reference to Figures 3.2 and 4.2, the following difference equations are used to model the AN/ASQ-81 magnetometer filter transfer functions. The input to the fixed high pass filter is called SIG(I), where I is the current data sample. The output of the fixed high pass filter, which is the input to the selectable high pass filter, is YO(I), and the output of the selectable high pass filter, the input to the selectable low pass filter, is called YPO(I). The output of the filter is called ASQ(I). (I-1) denotes a time delay of one sample, and so forth, and the symbol * denotes multiplication.

For the fixed high pass filter:

$$YO(I) = BFHP0 * SIG(I) + BFHP1 * SIG(I-1) + BFHP2 * SIG(I-2) + AFHP1 * YO(I-1) + AFHP2 * YO(I-2)$$

where:

$$BFHP0 = 0.9980499222938581$$

$$BFHP1 = -1.9960998445877161 \quad AFHP1 = 1.9960983216843922$$

$$BFHP2 = 0.998049922238581 \quad AFHP2 = -0.9961013674910398$$

For the selectable high pass filters:

$$XI(I) = ASHP1 * YO(I) + ASHP2 * XI(I-2) + ASHP3 * XI(I-3)$$

$$XII(I) = XI(I) + XI(I-2) - 2 * XI(I-1)$$

$$XIII(I) = XII(I) + ASHP4 * XIII(I-1) + ASHP5 * XIII(I-2)$$

$$XIV(I) = XIII(I) - 2 * XIII(I-1) + XIII(I-2)$$

$$XV(I) = XIV(I) + ASHP6 * XV(I-1) + ASHP7 * XV(I-2)$$

$$YPO(I) = XV(I) - 2 * XV(I-1) + XV(I-2)$$

For the low frequency cutoff at 0.04 HZ:

$$ASHP1 = 0.994713789347288 \quad ASHP2 = -0.9952196910157882$$

$$ASHP3 = 1.9952137256322473 \quad ASHP4 = 1.9962028201103847$$

$$ASHP5 = -0.9962082013015601 \quad ASHP6 = 1.9979855321466768$$

$$ASHP7 = -0.9979897681491607$$

For the low frequency cutoff at 0.06 HZ:

$$ASHP1 = 0.9920821277199399 \quad ASHP2 = -0.9928381306174365$$

$$ASHP3 = 1.9928247245317958 \quad ASHP4 = 1.9943056083414792$$

$$ASHP5 = -0.9943177045263504 \quad ASHP6 = 1.9969766455640039$$

$$ASHP7 = -0.9969861717656565$$

For the low frequency cutoff at 0.08 HZ:

$$ASHP1 = 0.9894580558875787 \quad ASHP2 = -0.9904622611337722$$

$$ASHP3 = 1.9904384565912725 \quad ASHP4 = 1.9924092959984783$$

$$ASHP5 = -0.9924307799269113 \quad ASHP6 = 1.9959666590811389$$

$$ASHP7 = -0.9959835860201267$$

For the low frequency cutoff at 0.10 HZ:

$$ASHP1 = 0.9869415560096681 \quad ASHP2 = -0.9880929619779491$$

$$ASHP3 = 1.9880549117849344 \quad ASHP4 = 1.9905139074397893$$

$$ASHP5 = -0.9905474442556225 \quad ASHP6 = 1.9949555824611562$$

$$ASHP7 = -0.9949820174650785$$

For the selectable low pass filters:

$$ASQ(I) = ASLP1 * ASQ(I-1) + ASLP2 * ASQ(I-2) + ASLP3 * ASQ(I-3)$$

$$+ ASLP4 * ASQ(I-4) + BSLP0 * YPO(I) + BSLP1 * YPO(I-1)$$

$$+ BSLP2 * YPO(I-2) + BSLP3 * YPO(I-3) + BSLP4 * YPO(I-4)$$

For the high frequency cutoff at 0.2 HZ:

BSLP0 = 0.0000000452616229

BSLP1 = 0.0000001810464917 ASLP1 = 3.9082436339591027

BSLP2 = 0.0000002715697375 ASLP2 = -5.7285022156249328

BSLP3 = 0.0000001810464917 ASLP3 = 3.7321935213310065

BSLP4 = 0.0000000452616229 ASLP4 = -0.9119356638511430

For the high frequency cutoff at 0.4 HZ:

BSLP0 = 0.0000006918001209

BSLP1 = 0.0000027672004837 ASLP1 = 3.8173771378993420

BSLP2 = 0.0000041508007256 ASLP2 = -5.4670046844062743

BSLP3 = 0.0000027672004837 ASLP3 = 3.4812554457127576

BSLP4 = 0.0000006918001209 ASLP4 = -0.8316389680077603

For the high frequency cutoff at 0.6 HZ:

BSLP0 = 0.0000033463317975

BSLP1 = 0.0000133853271900 ASLP1 = 3.7274299052305002

BSLP2 = 0.0000200779907850 ASLP2 = -5.2152772583906819

BSLP3 = 0.0000133853271900 ASLP3 = 3.2462216641520829

BSLP4 = 0.0000033463317975 ASLP4 = -0.7584278523006615

APPENDIX E

DIGITAL SOFTWARE FOR SIMULATION - DIRECT FORM (SINUSOIDS AS INPUT)

```

//HUETE JOB (1457,1106), ' ', CLASS=B
//EXEC FRTXCLGP
//FORT.SYSIN DD *
CCCCCCCCCCCC
THIS PROGRAM IS DESIGNED TO TEST THE ACTION OF THE PRELIMINARY
DIGITAL FILTER PROGRAM FOR THE ASQ-81 BY INTRODUCING A SINUSOID
AS THE INPUT SIGNAL TO THE FILTER
MGN3
SET UP ARRAYS. SIG() IS THE SIGNAL, ASQ() IS THE PROGRAM
OUTPUT, TRU() IS THE SIGNAL WITHIN THE FREQUENCY RANGE OF THE
PROGRAM
DIMENSION SIG(3000), ASQ(3000), TRU(3000), TIME(3000)
DIMENSION YQ(3000), YPO(3000)
REAL*8 DSEED
REAL*8 T, AFHP1, AFHP2, BFHP0, BFHP1, BFHP2, A, B, C, D, E, F
REAL*8 A1, B1, C1, D1, E1, F1, G1, H1, I1, J1, K1, L1
REAL*8 AA, BB, CC, DD, EE, FF, AA1, BB1, CC1, DD1, EE1, FF1, GG1, HH1, I11
REAL*8 JJ1, KK1, LL1
REAL*8 ASHP41, ASHP42, ASHP43, ASHP44, ASHP45, ASHP46
REAL*8 BSHP41, BSHP42, BSHP43, BSHP44, BSHP45, BSHP46
REAL*8 ASLP61, ASLP62, ASLP63, ASLP64
REAL*8 BSLP60, BSLP61, BSLP62, BSLP63, BSLP64
REAL*8 TITLA(12), HUETE ' ', Y OUT SI, GNAL ' ', 1457P ' ',
$8*
REAL*8 TITLB(12), HUETE ' ', OUTPUT S, IGNAL ' ', 1457P ' ',
$8*
REAL*8 TITLC(12), HUETE ' ', YPRI SIG, NAL ' ', 1457P ' ',
$8*
REAL LABEL / ' ' /
DATA PI/3.141592954/
DOUBLE PRECISION DSEED
DEFINE AND COMPUTE ALL COEFFICIENTS
TEN SAMPLES PER SECOND
T=1./10.
COEFFICIENTS FOR FIXED HIGH PASS FILTER
AFHP1=-((T**2/160.-2.)/(1.+T/8.+T**2/320.))
AFHP2=-((1.-T/8.+T**2/320.)/(1.+T/8.+T**2/320.))
BFHP0=(1.)/(1.+T/8.+T**2/320.))
BFHP1=-((2.)/(1.+T/8.+T**2/320.))
BFHP2=(1.)/(1.+T/8.+T**2/320.))
COEFFICIENTS FOR SELECTABLE HIGH PASS FILTER
IN THIS CASE, F(LOWER)=0.04 HZ
CCCCCCCCCCCC

```


C

A=12.52096/40.82834
 B=1./40.82834
 C=11.00999/45.28317
 D=1./45.28317
 E=7.41458/57.57668
 F=1./57.57668
 AI=1.+A*I/2.+B*(T**2)/4.
 BI=-2.+B*(T**2/2.)
 CI=1.-A*I/2.+B*(T**2)/4.
 DI=1.+C*I/2.+D*(T**2)/4.
 EI=-2.+D*(T**2)/2.
 FI=1.-C*I/2.+D*(T**2)/4.
 GI=1.+E*I/2.+F*(T**2)/4.
 HI=-2.+F*(T**2)/2.
 I1=1.-E*I/2.+F*(T**2)/4.

CCCC

CODE IS "ASHP41" MEANS "A1 COEFFICIENT FOR THE SELECTABLE HIGH
 PASS FILTER WITH LOWER LIMIT 0.04 HZ"

$$\text{ASHP41} = -(\text{G1} * (\text{A1} * \text{EI} + \text{B1} * \text{D1})) + (\text{H1} * \text{A1} * \text{D1}) / (\text{G1} * \text{A1} * \text{D1})$$

$$\$ (\text{G1} * \text{A1} * \text{D1})$$

$$\text{ASHP42} = -(\text{G1} * (\text{A1} * \text{FI} + \text{B1} * \text{EI} + \text{C1} * \text{D1})) + \text{H1} * (\text{A1} * \text{EI} + \text{B1} * \text{D1})) /$$

$$\$ / (\text{G1} * \text{A1} * \text{D1})$$

$$\text{ASHP43} = -(\text{G1} * (\text{B1} * \text{FI} + \text{C1} * \text{EI})) + \text{H1} * (\text{A1} * \text{FI} + \text{B1} * \text{EI} + \text{C1} * \text{D1})) + \text{I1} * (\text{A1} * \text{EI} + \text{B1} * \text{D1})) /$$

$$\$ / (\text{G1} * \text{A1} * \text{D1})$$

$$\text{ASHP44} = -(\text{G1} * \text{C1} * \text{FI} + \text{H1} * (\text{B1} * \text{FI} + \text{C1} * \text{EI})) + \text{I1} * (\text{A1} * \text{FI} + \text{B1} * \text{EI} + \text{C1} * \text{D1})) / (\text{G1} * \text{A1} * \text{D1})$$

$$\$ \text{A1} * \text{D1})$$

$$\text{ASHP45} = -(\text{H1} * \text{C1} * \text{FI} + \text{I1} * (\text{B1} * \text{FI} + \text{C1} * \text{EI})) / (\text{G1} * \text{A1} * \text{D1})$$

$$\text{ASHP46} = -(\text{I1} * \text{C1} * \text{FI}) / (\text{G1} * \text{A1} * \text{D1})$$

$$\text{BSHP40} = 1. / (\text{G1} * \text{A1} * \text{D1})$$

$$\text{BSHP41} = -6. / (\text{G1} * \text{A1} * \text{D1})$$

$$\text{BSHP42} = 15. / (\text{G1} * \text{A1} * \text{D1})$$

$$\text{BSHP43} = -20. / (\text{G1} * \text{A1} * \text{D1})$$

$$\text{BSHP44} = 15. / (\text{G1} * \text{A1} * \text{D1})$$

$$\text{BSHP45} = -6. / (\text{G1} * \text{A1} * \text{D1})$$

$$\text{BSHP46} = 1. / (\text{G1} * \text{A1} * \text{D1})$$

CCCC

COEFFICIENTS FOR SELECTABLE LOW PASS FILTER WITH UPPER FREQ.
 OF 0.6 HZ

AA=1./0.03452
 BB=0.35804/C.03492
 CC=1./0.03492
 DD=1./0.02779
 EE=0.20696/0.02779
 FF=1./0.02779
 AA1=AA*(T**2)/4.
 BB1=2.*AA1

APP00490
 APP00500
 APP00510
 APP00520
 APP00530
 APP00540
 APP00550
 APP00560
 APP00570
 APP00580
 APP00590
 APP00600
 APP00610
 APP00620
 APP00630
 APP00640
 APP00650
 APP00660
 APP00670
 APP00680
 APP00690
 APP00700
 APP00710
 APP00720
 APP00730
 APP00740
 APP00750
 APP00760
 APP00770
 APP00780
 APP00790
 APP00800
 APP00810
 APP00820
 APP00830
 APP00840
 APP00850
 APP00860
 APP00870
 APP00880
 APP00890
 APP00900
 APP00910
 APP00920
 APP00930
 APP00940
 APP00950
 APP00960


```

CC1=AA1
DD1=DD*(T**2)/4.
EE1=2.*DD1
FF1=DD1
GG1=(1.+BB*T/2.+CC*(T**2)/4.)
HH1=(1.+CC*(T**2)/2.)
II1=(1.-BB*T/2.+CC*(T**2)/4.)
JJ1=(1.+EE*T/2.+FF*(T**2)/4.)
KK1=(1.-EE*(T**2)/2.)
LL1=(1.-EE*T/2.+FF*(T**2)/4.)
ASLP61=-(GG1*KK1+HH1*JJ1)/(GG1*JJ1)
ASLP62=-(GG1*LL1+HH1*KK1+II1*JJ1)/(GG1*JJ1)
ASLP63=-(HH1*LL1+II1*KK1)/(GG1*JJ1)
ASLP64=-(II1*LL1)/(GG1*JJ1)
BSLP60=(AA1*DD1)/(GG1*JJ1)
BSLP61=(AA1*EE1+BB1*DD1)/(GG1*JJ1)
BSLP62=(AA1*FF1+BB1*EE1+CC1*DD1)/(GG1*JJ1)
BSLP63=(BB1*FF1+CC1*EE1)/(GG1*JJ1)
BSLP64=(CC1*FF1)/(GG1*JJ1)

FINISHED COMPUTING COEFFICIENTS, INITIALIZE STORAGE REGISTERS

DO 200 J=1,3000
  TRU(J)=0.
  YD(J)=0.
  YPO(J)=0.
  ASQ(J)=0.
  SIG(J)=0.
  200 CONTINUE

  STORAGE REGISTERS SET TO 0:
  SET UP EQUATIONS FOR FILTER

DO 100 I=1,3000
  I1=I-1
  I2=I-2
  I3=I-3
  I4=I-4
  I5=I-5
  I6=I-6
  IF(I1.LT.1) I1=1
  IF(I2.LT.1) I2=1
  IF(I3.LT.1) I3=1
  IF(I4.LT.1) I4=1
  IF(I5.LT.1) I5=1
  IF(I6.LT.1) I6=1
  ANOI=GGUBFS(DSEED)

```

CC C

CC CC C

C

APP01930
 APP01940
 APP01950
 APP01960
 APP01970
 APP01980
 APP01990
 APP02000
 APP02010
 APP02020
 APP02030
 APP02040
 APP02050

```

CALL DRAW(3000, TIME, ASQ, 0, 0, LABEL, TITLB, 0, 0, 0, 0, 0, 0, 5, 4, 1, LAST)
THIRD PLOT WILL BE A PLOT OF THE 'TRUE' SIGNAL VERSUS TIME
WITH SIGNAL ON THE X AXIS AND TIME ON THE Y AXIS
CALL DRAW(3000, TIME, YPO, 0, 0, LABEL, TITLC, 0, 0, 0, 0, 0, 0, 5, 4, 1, LAST)
FINISHED PLOTTING
STOP
END

```

C C C C C C C /*

APPENDIX F

DIGITAL SOFTWARE FOR SIMULATION - CASCADE FORM (SINUSOIDS AS INPUT)

```

//HUETE JOB (1457,1106), ' ', CLASS=B
//EXEC FRTXCLGP
//FORT.SYSIN DD *
C
C      MGN5 FORTRAN
C      THIS PROGRAM IS DESIGNED TO TEST THE ACTION OF THE PRELIMINARY
C      DIGITAL FILTER PROGRAM FOR THE ASQ-81 BY INTRODUCING SIMULATED
C      MAGNETIC SIGNALS INTO THE SYSTEM THROUGH THE USE OF SINUSOIDS
C      OF VARYING FREQUENCY, BOTH WITHIN AND OUTSIDE THE FREQUENCY
C      RANGE OF THE FILTER, WITH RANDOM NOISE ADDED
C
C      SET UP ARRAYS. SIG() IS THE SIGNAL, ASQ() IS THE PROGRAM
C      OUTPUT, TRU() IS THE SIGNAL WITHIN THE FREQUENCY RANGE OF THE
C      PROGRAM
C
C      DIMENSION SIG(3000), ASQ(3000), TRU(3000), TIME(3000)
C      DIMENSION VO(3000), YPO(3000)
C      DIMENSION XI(3000), XII(3000), XIV(3000), XV(3000)
C      REAL*8 DSEEC
C      REAL*8 T, AFHP1, AFHP2, BFHP0, BFHP1, BFHP2, A, B, C, D, E, F
C      REAL*8 A1, B1, C1, D1, E1, F1, G1, H1, I1, J1, K1, L1
C      REAL*8 AA, BB, CC, DD, EE, FF, AA1, BB1, CC1, DD1, EE1, FF1, GG1, HH1, I11
C      REAL*8 JJ1, KK1, LL1
C      REAL*8 ASHP41, ASHP42, ASHP43, ASHP44, ASHP45, ASHP46, ASHP47
C      REAL*8 ASLP61, ASLP62, ASLP63, ASLP64
C      REAL*8 BSLP60, BSLP61, BSLP62, BSLP63, BSLP64
C      REAL*8 TITLA(12), HUETE
C      $8* TITLB(12), HUETE
C      $8* TITLC(12), HUETE
C      $8* TITLD(12), HUETE
C      $8* TITLE(12), HUETE
C      REAL LABEL, /
C      DATA PI/3.141592954/
C      DOUBLE PRECISION DSEED
C
C      DEFINE AND COMPUTE ALL COEFFICIENTS
C      TEN SAMPLES PER SECOND
C
C      T=1./10.
C
C      COEFFICIENTS FOR FIXED HIGH PASS FILTER
C
C      AFHP1=-((T**2/160.-2.)/(1.+T/8.+T**2/320.))
C      AFHP2=-((1.-T/8.+T**2/320.)/(1.+T/8.+T**2/320.))

```



```

BFHP0=(1./((1.+T/8.+T**2/320.)),
BFHP1=-2./((1.+T/8.+T**2/320.)),
BFHP2=(1./((1.+T/8.+T**2/320.)))
COEFFICIENTS FOR SELECTABLE HIGH PASS FILTER
IN THIS CASE, F(LOWER)=0.04 HZ
A=12.52096/40.82834
B=1./40.82834
C=11.00999/45.28317
D=1./45.28317
E=7.41498/57.57668
F=1./57.57668
A1=1.+A*T/2.+B*(T**2)/4.
B1=-2.+B*(T**2/2.)
C1=1.-A*T/2.+B*(T**2)/4.
D1=1.+C*T/2.+D*(T**2)/4.
E1=-2.+D*(T**2)/2.
F1=1.-C*T/2.+D*(T**2)/4.
G1=1.+E*T/2.+F*(T**2)/4.
H1=-2.+F*(T**2)/2.
I1=1.-E*T/2.+F*(T**2)/4.
CODE IS "ASHP41" MEANS "A1 COEFFICIENT FOR THE SELECTABLE HIGH
PASS FILTER WITH LOWER LIMIT 0.04 HZ"
ASHP41=1./((A1*D1*G1))
ASHP42=-C1/A1)
ASHP43=-B1/A1)
ASHP44=-E1/D1)
ASHP45=-F1/D1)
ASHP46=-H1/G1)
ASHP47=-I1/G1)
COEFFICIENTS FOR SELECTABLE LOW PASS FILTER WITH UPPER FREQ.
OF 0.6 HZ
AA=1./0.03492
BB=0.35804/C.03492
CC=1./0.03492
DD=1./0.02779
EE=0.20696/C.02779
FF=1./0.02779
AA1=AA*(T**2)/4.
BB1=2.*AA1
CC1=AA1
DD1=DD*(T**2)/4.
EE1=2.*DD1

```

CCCC

CCCC

CCCC


```

APP03040
APP03050
APP03060
APP03070
APP03080
APP03090
APP03100
APP03110
APP03120
APP03130
APP03140
APP03150
APP03160
APP03170
APP03180
APP03190
APP03200
APP03210
APP03220
APP03230
APP03240
APP03250
APP03260
APP03270
APP03280
APP03290
APP03300
APP03310
APP03320
APP03330
APP03340
APP03350
APP03360
APP03370
APP03380
APP03390
APP03400
APP03410
APP03420
APP03430
APP03440
APP03450
APP03460
APP03470
APP03480
APP03490
APP03500
APP03510

```

```

FF1=DD1+BB*T/2.+CC*(T**2)/4.)
GG1=(-2.+CC*(T**2)/2.)
HH1=(-2.+BB*T/2.+CC*(T**2)/4.)
II1=(1.-BB*T/2.+CC*(T**2)/4.)
JJ1=(-2.+EE*(T**2)/2.)
KK1=(1.-EE*T/2.+FF*(T**2)/4.)
LL1=(1.-GG*KK1+HH*JJ1)/(GG1*JJ1)
ASLP61=-((GG1*LL1+HH*KK1+II*JJ1)/(GG1*JJ1)
ASLP62=-((HH1*LL1+II*KK1)/(GG1*JJ1)
ASLP63=-((II*LL1)/(GG1*JJ1)
ASLP64=-((AA1*DD1)/(GG1*JJ1)
BSLP60=(AA1*EE1+BB1*DD1)/(GG1*JJ1)
BSLP61=(AA1*EE1+BB1*DD1)/(GG1*JJ1)
BSLP62=(AA1*FF1+BB1*EE1+CC1*DD1)/(GG1*JJ1)
BSLP63=(BB1*FF1+CC1*EE1)/(GG1*JJ1)
BSLP64=(CC1*FF1)/(GG1*JJ1)

```

FINISHED COMPUTING COEFFICIENTS, INITIALIZE STORAGE REGISTERS

```

DO 200 J=1,3000
TRU(J)=0.
VO(J)=0.
YPO(J)=0.
ASQ(J)=0.
SIG(J)=0.
CONTINUE

```

200

STORAGE REGISTERS SET TO 0, EQUATIONS FOR SIMULATED SIGNAL

```

DSEED = 1456.
PHI1=GGUBFS(DSEED)*2.*PI
PHI2=GGUBFS(DSEED)*2.*PI
DO 100 I=1,3000
I1=I-1
I2=I-2
I3=I-3
I4=I-4
I5=I-5
I6=I-6
IF(I1.LT.1) I1=1
IF(I2.LT.1) I2=1
IF(I3.LT.1) I3=1
IF(I4.LT.1) I4=1
IF(I5.LT.1) I5=1
IF(I6.LT.1) I6=1
TRU(I)=SIN(0.02*PI*FLOAT(I)+PHI1)
ANOI=GGUBFS(DSEED)

```

C

APP04000
APP04010
APP04020
APP04030
APP04040
APP04050
APP04060
APP04070

CALL DRAW(3000, TIME, SIG, 0, 0, LABEL, TITLA, 0, 0, 0, 0, 0, 5, 4, 1, LAST)
CALL DRAW(3000, TIME, TRU, 0, 0, LABEL, TITLC, 0, 0, 0, 0, 0, 5, 4, 1, LAST)

FINISHED PLOTTING

STOP
END

C
C
C

/*

APPENDIX G

DIGITAL SOFTWARE FOR COMPUTATION OF SYSTEM AMPLITUDE VERSUS FREQUENCY

```

//HUEETE JOB (1457,1106), ' ', CLASS=B
//EXEC FRTXCLGP
//FORT.SYSIN DD *
CCCCCCCCCCCC
      MGN4 FORTRAN
THIS PROGRAM IS DESIGNED TO INPUT VARIOUS FREQUENCIES INTO THE
DIGITAL FILTER PROGRAM FOR THE ASQ-81 AND OBTAIN THE DB LOSS
CHARACTERISTIC FOR COMPARISON WITH MEASURED DB LOSSES FOR THE
ASQ 81 MAGNETOMETER. A SINGLE FREQUENCY SIGNAL WILL BE INPUTED
AND THE RMS OUTPUT DIVIDED BY THE RMS INPUT TO DETERMINE
ATTENUATION
SET UP ARRAYS. SIG() IS THE SIGNAL, ASQ() IS THE PROGRAM
OUTPUT, TRU() IS THE SIGNAL WITHIN THE FREQUENCY RANGE OF THE
PROGRAM
      DIMENSION SIG(3000), ASQ(3000), TRU(3000), TIME(3000)
      DIMENSION YG(3000), YPD(3000), FREQ(20)
      DIMENSION XI(3000), XII(3000), XIV(3000), XV(3000)
      REAL*8 DSEED
      REAL*8 T, AFHP1, AFHP2, BFHP0, BFHP1, BFHP2, A, B, C, D, E, F
      REAL*8 AI, BI, CI, DI, EI, FI, GI, HI, II, JI, KI, LI
      REAL*8 AA, BB, CC, DD, EE, FF, AA1, BB1, CC1, DD1, EE1, FF1, GG1, HH1, II1
      REAL*8 JJ1, KK1, LL1
      REAL*8 ASHP41, ASHP42, ASHP43, ASHP44, ASHP45, ASHP46, ASHP47
      REAL*8 ASLP61, ASLP62, ASLP63, ASLP64
      REAL*8 BSLP60, BSLP61, BSLP62, BSLP63, BSLP64
      DATA PI/3.141592954/
      DOUBLE PRECISION DSEED, SUMSQ, SMSQT, RATIO
      DEFINE AND COMPUTE ALL COEFFICIENTS
      TEN SAMPLES PER SECOND
      T=1./10.
      COEFFICIENTS FOR FIXED HIGH PASS FILTER
      AFHP1=-((T**2/160.-2.)/(1.+T/8.+T**2/320.))
      AFHP2=-((1.-T/8.+T**2/320.)/(1.+T/8.+T**2/320.))
      BFHP0=(1./(1.+T/8.+T**2/320.))
      BFHP1=-((2.)/(1.+T/8.+T**2/320.))
      BFHP2=(1./(1.+T/8.+T**2/320.))
      COEFFICIENTS FOR SELECTABLE HIGH PASS FILTER
      IN THIS CASE, F(LOWER)=0.04 HZ
      A=12.52096/40.82834
      B=1./40.82834
      C=11.00999/45.28317
      D=1./45.28317

```



```

E=7.41458/57.57668
F=1./57.57668
A1=1.+A*T/2.+B*(T**2)/4.
B1=-2.+B*(T**2/2.)+B*(T**2)/4.
C1=1.-A*T/2.+B*(T**2)/4.
D1=1.+C*T/2.+D*(T**2)/4.
E1=-2.+D*(T**2)/2.+D*(T**2)/4.
F1=1.-C*T/2.+D*(T**2)/4.
G1=1.+E*T/2.+F*(T**2)/4.
H1=-2.+F*(T**2)/2.
I1=1.-E*T/2.+F*(T**2)/4.

CODE IS "ASHP41" MEANS "A1 COEFFICIENT FOR THE SELECTABLE HIGH
PASS FILTER WITH LOWER LIMIT 0.04 HZ"

ASHP41=1./((A1*D1*G1))
ASHP42=- (C1/A1)
ASHP43=- (B1/A1)
ASHP44=- (E1/D1)
ASHP45=- (F1/D1)
ASHP46=- (H1/G1)
ASHP47=- (I1/G1)

COEFFICIENTS FOR SELECTABLE LOW PASS FILTER WITH UPPER FREQ.
OF 0.6 HZ

AA=1./0.03492
BB=0.35804/C.03492
CC=1./0.03492
DD=1./0.02779
EE=0.20696/0.02779
FF=1./0.02779
AA1=AA*(T**2)/4.
BB1=2.*AA1
CC1=AA1
DD1=DD*(T**2)/4.
EE1=2.*DD1
FF1=DD1
GG1=(1.+BB*T/2.+CC*(T**2)/4.)
HH1=(-2.+CC*(T**2)/2.)
II1=(1.-BB*T/2.+CC*(T**2)/4.)
JJ1=(1.+EE*T/2.+FF*(T**2)/4.)
KK1=(-2.+FF*(T**2)/2.)
LL1=(1.-EE*T/2.+FF*(T**2)/4.)
ASLP61=- (GG1*KK1+HH1*JJ1)/(GG1*JJ1)
ASLP62=- (GG1*LL1+HH1*KK1+II1*JJ1)/(GG1*JJ1)
ASLP63=- (HH1*LL1+II1*KK1)/(GG1*JJ1)
ASLP64=- (II1*LL1)/(GG1*JJ1)

```

CCCC

CCCC


```

C
C
C
BSLP60=(AA1*DD1)/(GG1*JJ1)
BSLP61=(AA1*EE1+BB1*DD1)/(GG1*JJ1)
BSLP62=(AA1*FF1+BB1*EE1+CC1*DD1)/(GG1*JJ1)
BSLP63=(AA1*FF1+CC1*EE1)/(GG1*JJ1)
BSLP64=(CC1*FF1)/(GG1*JJ1)
FINISHED COMPUTING COEFFICIENTS, INITIALIZE STORAGE REGISTERS
DO 200 J=1,3000
  TRU(J)=0.
  YO(J)=0.
  YPO(J)=0.
  ASQ(J)=0.
  SIG(J)=0.
  CONTINUE
200
C
C
C
STORAGE REGISTERS SET TO 0, EQUATIONS FOR SIMULATED SIGNAL
DSEED = 1456.
PHI1=GGUBFS(DSEED)*2.*PI
PHI2=GGUBFS(DSEED)*2.*PI
FREQ(1)=0.01
FREQ(2)=0.02
FREQ(3)=0.03
FREQ(4)=0.04
FREQ(5)=0.05
FREQ(6)=0.06
FREQ(7)=0.07
FREQ(8)=0.08
FREQ(9)=0.09
FREQ(10)=0.10
FREQ(11)=0.20
FREQ(12)=0.30
FREQ(13)=0.40
FREQ(14)=0.50
FREQ(15)=0.60
FREQ(16)=0.70
FREQ(17)=0.80
FREQ(18)=0.90
FREQ(19)=1.0
FREQ(20)=0.15
DO 300 I=1,20
  DO 100 I=1,3000
    I1=I-1
    I2=I-2
    I3=I-3
    I4=I-4
    I5=I-5
APP05060
APP05070
APP05080
APP05090
APP05100
APP05110
APP05120
APP05130
APP05140
APP05150
APP05160
APP05170
APP05180
APP05190
APP05200
APP05210
APP05220
APP05230
APP05240
APP05250
APP05260
APP05270
APP05280
APP05290
APP05300
APP05310
APP05320
APP05330
APP05340
APP05350
APP05360
APP05370
APP05380
APP05390
APP05400
APP05410
APP05420
APP05430
APP05440
APP05450
APP05460
APP05470
APP05480
APP05490
APP05500
APP05510
APP05520
APP05530

```



```

I6=I-6
IF(I1.LT.I1) I1=I
IF(I2.LT.I1) I2=I
IF(I3.LT.I1) I3=I
IF(I4.LT.I1) I4=I
IF(I5.LT.I1) I5=I
IF(I6.LT.I1) I6=I
TRU(I)=SIN(0.2*PI*FREQ(I1)*FLOAT(I1))
SIG(I)=TRU(I)
TIME(I)=FLOAT(I)/600.

THE FIRST STAGE OF THE FILTER PROGRAM IS COMMENTED OUT BECAUSE IN
THIS VERSION OF THE PROGRAM THE FIXED HIGH PASS FILTER IS NOT
INCLUDED SO AS TO ENABLE COMPARISON WITH DATA FURNISHED BY
TEXAS INSTRUMENTS, INC. BY REMOVING THE COMMENT CHARACTERS, AND
COMMENTING OUT THE STEP FOLLOWING THEM ( YO(I)=SIG(I) ), THE
PROGRAM CAN BE MADE TO INCLUDE THE FIXED HIGH PASS FILTER

YO(I)=BFHP0*SIG(I)+BFHP1*SIG(I1)+BFHP2*SIG(I2)+AFHP1*YO(I1)
$+AFHP2*YO(I2)
YO(I)=SIG(I)
XI(I)=ASHP41*YO(I)+ASHP42*XI(I2)+ASHP43*XI(I1)
XI(I1)=XI(I1)+XI(I2)-2.*XI(I1)
XI(I1)=XI(I1)+ASHP44*XI(I1)+ASHP45*XI(I2)
XIV(I)=XI(I1)-2.*XI(I1)+XI(I2)
XV(I)=XIV(I)+ASHP46*XV(I1)+ASHP47*XV(I2)
YPO(I)=XV(I1)+XV(I2)-2.*XV(I1)
GP1=ASLP61*ASQ(I1)+ASLP62*ASQ(I2)+ASLP63*ASQ(I3)+ASLP64*ASQ(I4)
GP2=BSLP60*YPO(I)+BSLP61*YPO(I1)+BSLP62*YPO(I2)
$+BSLP63*YPO(I3)+BSLP64*YPO(I4)
ASQ(I)=GP1+GP2

100 CONTINUE

THE FOLLOWING SECTION COMPUTES THE AVERAGE VALUES OF THE OUTPUT
AND CONVERTS TO DB ATTENUATION

SUMSQ=0.0
SMSQT=0.0
DO 301 JJ=2000,3000
SUMSQ=SUMSQ+(ASQ(JJ)**2)
SMSQT=SMSQT+(TRU(JJ)**2)
301 CONTINUE
RATIO=SUMSQ/SMSQT
DBLOSS=10.*DLG10(RATIO)
WRITE(6,4001)FREQ(I1),DBLOSS
4001 FORMAT(IX,' FREQUENCY = ',F10.2,' DB LOSS =',F10.5)

```


C
C
C
FINISHED PLOTTING
300 CONTINUE
STOP
END
/*

APP06020
APP06030
APP06040
APP06050
APP06060
APP06070
APP06080

DIGITAL SOFTWARE FOR SIMULATION (ANDERSON FUNCTIONS AS INPUT)

UUUU UU

APP06590
APP06600
APP06610
APP06620
APP06630
APP06640
APP06650
APP06660
APP06670
APP06680
APP06690
APP06700
APP06710
APP06720
APP06730
APP06740
APP06750
APP06760
APP06770
APP06780
APP06790
APP06800
APP06810
APP06820
APP06830
APP06840
APP06850
APP06860
APP06870
APP06880
APP06890
APP06900
APP06910
APP06920
APP06930
APP06940
APP06950
APP06960
APP06970
APP06980
APP06990
APP07000
APP07010
APP07020
APP07030
APP07040
APP07050
APP07060

BFHP2=(1./ (1.+T/8.+T**2/320.))
COEFFICIENTS FOR SELECTABLE HIGH PASS FILTER
IN THIS CASE, F(LOWER)=0.04 HZ

A=12.52096/40.82834
B=1./40.82834
C=11.00999/45.28317
D=1./45.28317
E=7.41459/57.57668
F=1./57.57668
A1=1.+A*T/2.+B*(T**2)/4.
B1=-2.+B*(T**2/2.)
C1=1.-A*T/2.+B*(T**2)/4.
D1=1.+C*T/2.+D*(T**2)/4.
E1=-2.+D*(T**2)/2.
F1=1.-C*T/2.+D*(T**2)/4.
G1=1.+E*T/2.+F*(T**2)/4.
H1=-2.+F*(T**2)/2.
I1=1.-E*T/2.+F*(T**2)/4.

CODE IS "ASHP41" MEANS "A1 COEFFICIENT FOR THE SELECTABLE HIGH
PASS FILTER WITH LOWER LIMIT 0.04 HZ"

ASHP41=1./ (A1*D1*G1)
ASHP42=- (C1/A1)
ASHP43=- (B1/A1)
ASHP44=- (E1/D1)
ASHP45=- (F1/D1)
ASHP46=- (H1/G1)
ASHP47=- (I1/G1)

COEFFICIENTS FOR SELECTABLE LOW PASS FILTER WITH UPPER FREQ.
OF 0.6 HZ

AA=1./0.03492
BB=0.35804/C.03492
CC=1./0.03492
DD=1./0.02779
EE=0.20696/C.02779
FF=1./0.02779
AA1=AA*(T**2)/4.
BB1=2.*AA1
CC1=AA1
DD1=DD*(T**2)/4.
EE1=2.*DD1
FF1=DD1
GG1=(1.+BB*T/2.+CC*(T**2)/4.)

APP07070
APP07080
APP07090
APP07100
APP07110
APP07120
APP07130
APP07140
APP07150
APP07160
APP07170
APP07180
APP07190
APP07200
APP07210
APP07220
APP07230
APP07240
APP07250
APP07260
APP07270
APP07280
APP07290
APP07300
APP07310
APP07320
APP07330
APP07340
APP07350
APP07360
APP07370
APP07380
APP07390
APP07400
APP07410
APP07420
APP07430
APP07440
APP07450
APP07460
APP07470
APP07480
APP07490
APP07500
APP07510
APP07520
APP07530
APP07540

```

HH1=(-2.+CC*(T**2)/2.)
II1=(1.-BB*T/2.+CC*(T**2)/4.)
JJ1=(1.+EE*T/2.+FF*(T**2)/4.)
KK1=(-2.+FF*(T**2)/2.)
LL1=(1.-EE*T/2.+FF*(T**2)/4.)
ASLP61=-((GG1*KK1+HH1*JJ1)/(GG1*JJ1))
ASLP62=-((GG1*LL1+HH1*KK1+II1*JJ1)/(GG1*JJ1))
ASLP63=-((HH1*LL1+II1*KK1)/(GG1*JJ1))
ASLP64=-((II1*LL1)/(GG1*JJ1))
BSLP60=(AA1*DD1)/(GG1*JJ1)
BSLP61=(AA1*EE1+BB1*DD1)/(GG1*JJ1)
BSLP62=(AA1*FF1+BB1*EE1+CC1*DD1)/(GG1*JJ1)
BSLP63=(BB1*FF1+CC1*EE1)/(GG1*JJ1)
BSLP64=(CC1*FF1)/(GG1*JJ1)

```

FINISHED COMPUTING COEFFICIENTS, INITIALIZE STORAGE REGISTERS

```

DO 200 J=1,3000
TRU(J)=0.
YQ(J)=0.
YPO(J)=0.
ASQ(J)=0.
SIG(J)=0.
CONTINUE

```

200

STORAGE REGISTERS SET TO 0, EQUATIONS FOR SIMULATED SIGNAL

```

DO 100 I=1,2400
I1=I-1
I2=I-2
I3=I-3
I4=I-4
I5=I-5
I6=I-6
IF(I1.LT.1) I1=1
IF(I2.LT.1) I2=1
IF(I3.LT.1) I3=1
IF(I4.LT.1) I4=1
IF(I5.LT.1) I5=1
IF(I6.LT.1) I6=1

```

THIS SIGNAL STATEMENT INPUTS A 180 KNOTS AIRCRAFT AT A CPA RANGE OF 400 FEET. THIS IS TO TEST THE OPERATION OF THE SELCETABLE LOW PASS FILTER SECOND ANDERSON FUNCTION 8 HZ FILTER

```

TIME(I)=FLOAT(I)/480.
BETA=18000.*(TIME(I)-2.5)/400.

```

CCC

CCC

CCCCC

[illegible]

APPENDIX I DIGITAL FILTERING SOFTWARE

```

//HUETE JOB (1457,0165),*HUETE      SMC 2740*,CLASS=G
//*MAIN LINES=(65)
//EXEC FRTXCLGP,PARM.LKED='LIST,MAP,XREF',REGION.GO=2048K
//FORT.SYSIN DD *
C      INTEGER*2 IN(16)
C      ARRAY *IN* IS USED IN READING DATA FROM TAPE
C      COMPLEX*8 X(8192)
C      REAL*4 ZZ(8192),YV(8192),XXP(8192)
C      THE ABOVE COMPLEX*8 ARRAYS ARE USED TO ORDER INPUT DATA AND
C      INITIALLY REPRESENT VOLTAGE - TIME SERIES INFORMATION.
C      THE NEXT THREE LINES ARE ARRAYS NEEDED FOR DATA TAPE READING AND
C      CONVERSION TO TOTAL FIELD FLUCTUATION TIME SERIES
C      DIMENSION TIME(8192),FREQ(8192),WORK(24576),FRQ2(8192)
C      DIMENSION ZX1(8192),ZY1(8192)
C      DIMENSION ZZX1(24576),ZZY1(24576),ZZV1(24576)
C      THE FOLLOWING LINES CONTAIN ARRAYS NEEDED FOR SIGNAL INPUT TO THE
C      FILTER, SIGNAL PROCESSING WITHIN THE FILTER, AND COEFFICIENTS
C      USED BY THE FILTER PROGRAM
C      DIMENSION TIME2(24576),OUTFLD(24576),CLFLD(24576)
C      REAL*8 AH(4),BH(4),CH(4),DH(4),EH(4),FH(4)
C      REAL*8 AL(3),BL(3),CL(3),DL(3),EL(3),FL(3),FRQH(4),FRQL(3)
C      REAL*8 T,AFHP1,AFHP2,BFHP1,BFHP2,AL,B1,C1,D1,E1,F1,G1,H1
C      REAL*8 I1,J1,K1,L1,ASHP1,ASHP2,ASHP3,ASHP4,ASHP5,ASHP6,ASHP7
C      REAL*8 ASLP1,ASLP2,ASLP3,ASLP4,BSLP1,BSLP2,BSLP3,BSLP4
C      REAL*8 SIG2,SIG1,SIG,YO2,YO1,YO,XI2,XI1,XI,XI11,XI11,XV2
C      REAL*8 XV1,XV,YPO4,YPO3,YPO2,YPO1,YPO,OUTFD4,OUTFD3,OUTFD2
C      THE *Z* ARRAYS REPRESENT FREQUENCY DOMAIN (FF TRANSFORMED)
C      MAGNITUDE DATA AND ARE EVENTUALLY CONVERTED TO POWER SPECTRAL
C      DENSITY INFORMATION. ZX1,ZY1,ZV1,AND ZT1 REPRESENT MAGNITUDE
C      VALUES.
C      THE NEXT LINES CONTAIN CONSTANTS AND ARRAYS USED IN PLOTTING
C      THE OUTPUT
C      INTEGER K,I4,I5,Q
C      REAL SUMX,SUMY,SUMZ,AVE1
C      REAL CONSTX
C      INTEGER*4 ITB(12)/12*0/
C      REAL*4 RTB(28)/28*0.0/
C      REAL ALAB(4)
C      REAL*8 TTITLE(12)
C      EQUIVALENCE(TITLE(1),RTB(5))

```



```

C      SET VARIABLES EQUAL TO ZERO
DATA XX/8192*(0.,0.)/
DATA ZZ,YY/16384*0./
DATA ZX,1/8192*0./
DATA TIME,FREQ/16384*0./
K=0
I4=1
I5=1
CONSTX=0.0
SUMX=0.0
SUMY=0.0
SUMZ=0.0
AVE1=0.0
XORIGP=0.0
XMAXP=0.0

C      SET STORAGE REGISTERS TO ZERO.  STORAGE REGISTERS ARE USED VICE
INTERMEDIATE OUTPUT ARRAYS IN ORDER TO CUT DOWN THE AMOUNT OF
ARRAY STORAGE REQUIRED BY THE PROGRAM AND TO RETAIN "MEMORY"
OF PREVIOUS VALUES FOR COMPUTATIONAL USE IN ORDER TO ELIMINATE
THE "START UP" LAG OF THE OUTPUT VALUES
SIG2=0.0
SIG1=0.0
YD2=0.0
YD1=0.0
XI2=0.0
XI1=0.0
XI1I2=0.0
XI1I1=0.0
XV2=0.0
XV1=0.0
VP04=0.0
VP03=0.0
VP02=0.0
VP01=0.0
OUTFD4=0.0
OUTFD3=0.0
OUTFD2=0.0
OUTFD1=0.0

C      THE FOLLOWING SEVERAL STEPS WOULD BE USED IF THE INPUT TO THE
FILTER PROGRAM WERE THREE MUTUALLY PERPENDICULAR COIL SENSORS.
SINCE THE INPUT IS A SINGLE COIL SENSOR ORIENTED ALONG THE
EARTH'S FIELD, THESE STEPS ARE NOT NECESSARY, BUT ARE RETAINED
AS REFERENCE

```

```

APP08660
APP08670
APP08680
APP08690
APP08700
APP08710
APP08720
APP08730
APP08740
APP08750
APP08760
APP08770
APP08780
APP08790
APP08800
APP08810
APP08820
APP08830
APP08840
APP08850
APP08860
APP08870
APP08880
APP08890
APP08900
APP08910
APP08920
APP08930
APP08940
APP08950
APP08960
APP08970
APP08980
APP08990
APP09000
APP09010
APP09020
APP09030
APP09040
APP09050
APP09060
APP09070
APP09080
APP09090
APP09100
APP09110
APP09120
APP09130

```



```

C C C C C C C C
200 WRITE(6,200)IRR,IREC
FORMAT(10X,'IRR=',I6,5X,'IREC=',I6,/)
THE FOLLOWING SECTION GENERATES THE TIME AND FREQUENCY
ARRAYS AND NORMALIZES THE INPUT PCM DATA TO VOLTAGE FORM
IN PREPARATION FOR FAST FOURIER TRANSFORM TO THE FREQUENCY
DOMAIN.
N=8192
FN=FLOAT(N)
DELTAT=1./64.
DELTAF=1./((FN*DELTAT)
DO 20 J=1,N
TIME(J)=DELTAT*FLOAT(J)
FREQ(J)=DELTAF*FLOAT(J)
XX(J)=(XX(J)-2048.)*5./2048.
XX(J)=REAL(XX(J))
XXP(J)=XX(J)
YY(J)=(YY(J)-2048.)*5./2048.
ZZ(J)=(ZZ(J)-2048.)*5./2048.
IN THIS USE OF THE PROGRAM,
'XX' IS THE COIL ANTENNA DATA, 'YY' IS THE ASQ-81 DATA
'ZZ' IS THE SCHEDULED COIL DATA, AND 'TF' IS THE
TOTAL GEOMAGNETIC FIELD VECTOR.
IF THREE MUTUALLY PERPENDICULAR COIL SENSORS ARE USED, THIS
WILL NOT BE TRUE. SEE REFERENCE 9 FOR HOW TO HANDLE THIS
C C C C C C C C C C
20 CONTINUE
DO 21 J=1,N
FRQ2(J)=ALOG10(FREQ(J))
21 CONTINUE
NEXT FOUR STATEMENTS PERFORM AN FFT ON THE INPUT
TIME SERIES DATA. SEE THE WRITEUP ON 'FOURT' FOR
FURTHER INFORMATION.
CALL FOURT(XX,N,1,-1.0,WORK)
THE NEXT BLOCK OF STATEMENTS APPLY THE SYSTEM (VOLTAGE TO
B-FIELD) TRANSFER FUNCTION TO THE TRANSFORMED FREQUENCY
DOMAIN DATA. THIS BLOCK ENDS AT STATEMENT 9.
THE TRANSFER FUNCTION CONVERTS VOLTS TO NANOTESLAS (GAMMAS).
***WARNING*** THIS TRANSFER FUNCTIONS YIELDS AN INACCURATE
PHASE. USE A DIFFERENT TRANSFER FUNCTION IF PHASE INFORMATION
NEEDED.
DO 9 L=1,N
FRQ=FREQ(L)
IF(FRQ.LE.25.)GO TO 1
XX(L)=XX(L)/28.
GO TO 8
1 IF(FRQ.LE.15.)GO TO 2
XX(L)=XX(L)/(105.5-3.14*FRQ)
APP09620
APP09630
APP09640
APP09650
APP09660
APP09670
APP09680
APP09690
APP09700
APP09710
APP09720
APP09730
APP09740
APP09750
APP09760
APP09770
APP09780
APP09790
APP09800
APP09810
APP09820
APP09830
APP09840
APP09850
APP09860
APP09870
APP09880
APP09890
APP09900
APP09910
APP09920
APP09930
APP09940
APP09950
APP09960
APP09970
APP09980
APP09990
APP10000
APP10010
APP10020
APP10030
APP10040
APP10050
APP10060
APP10070
APP10080
APP10090

```



```

      GO TO 8
      2 IF (FRQ.LE.10.)GO TO 3
      XX(L)=XX(L)/(5.958*FRQ-30.97)
      GO TO 8
      3 IF (FRQ.LE.7.5)GO TO 4
      XX(L)=XX(L)/(3.492*FRQ-6.31)
      GO TO 8
      4 IF (FRQ.LE.5.)GO TO 5
      XX(L)=XX(L)/(2.6311*FRQ+0.14667)
      GO TO 8
      5 IF (FRQ.LE.3.)GO TO 6
      XX(L)=XX(L)/(2.6311*FRQ+0.14667)
      GO TO 8
      6 XX(L)=XX(L)/(2.72*FRQ)
      GO TO 8
      8 CONTINUE
      9 CALL FOURT(XX,N,1,1,1,WORK)
      DO 57 J=1,N
      XX(J)=XX(J)/FN
      57 CONTINUE
      WRITE(6,600)(XX(I),I=1,100)
      FORMAT(1X,F20.4,4X,F20.4)
      C 600 THE FOLLOWING BLOCK TAKES THE MAGNITUDE OF THE COMPLEX VALUES
      C
      DO 56 I3=1,N
      ZX1(I3)=CABS(XX(I3))
      56 CONTINUE
      IF(K.NE.0) GO TO 36
      DO 66 IS=8048,8192
      SUMX=ZX1(IS)+SUMX
      66 CONTINUE
      CONSTX=SUMX/144.
      DO 67 IS=1,8192
      ZX1(I4)=ZX1(IS)
      I4=I4+1
      67 CONTINUE
      36 GO TO 37
      CONTINUE
      SUMX=0.0
      DO 68 IS=1,144
      SUMX=ZX1(IS)+SUMX
      68 CONTINUE
      AVE1=SUMX/144.
      DO 69 IS=1,8192
      ZX1(I4)=ZX1(IS)+(CONSTX-AVE1)
      I4=I4+1
      69 CONTINUE
      37 CONTINUE

```

```

APP1 0100
APP1 0110
APP1 0120
APP1 0130
APP1 0140
APP1 0150
APP1 0160
APP1 0170
APP1 0180
APP1 0190
APP1 0200
APP1 0210
APP1 0220
APP1 0230
APP1 0240
APP1 0250
APP1 0260
APP1 0270
APP1 0280
APP1 0290
APP1 0300
APP1 0310
APP1 0320
APP1 0330
APP1 0340
APP1 0350
APP1 0360
APP1 0370
APP1 0380
APP1 0390
APP1 0400
APP1 0410
APP1 0420
APP1 0430
APP1 0440
APP1 0450
APP1 0460
APP1 0470
APP1 0480
APP1 0490
APP1 0500
APP1 0510
APP1 0520
APP1 0530
APP1 0540
APP1 0550
APP1 0560
APP1 0570

```



```

DO 91 I3=1, E192
ZZY1(I5)=YY(I3)*(1./7.5)
ZZV1(I5)=ZZ(I3)*10.
CLFLD(I5)=XXP(I3)
TIME2(I5)=(DELTA*FLOAT(I3))+(128.0*FLOAT(K))+XORIGP
I5=I5+1
91 CONTINUE
K=K+1
70 CONTINUE

      REMOVE DC VALUE OF OUTPUT TIME SERIES DATA IN ORDER TO PLOT
      FLUCTUATIONS

      SUMTT=0.0
      SUMTX=0.0
DO 223 KJ=1,24576
      SUMTX=SUMTX+CLFLD(KJ)
      SUMTT=SUMTT+ZZX1(KJ)
223 CONTINUE
      AVG=SUMTT/24576.
      AVX=SUMTX/24576.
DO 222 JK=1,24576
      CLFLD(JK)=CLFLD(JK)-AVX
      ZZX1(JK)=ZZX1(JK)-AVG
222 CONTINUE

      THIS NEXT SECTION IS USED TO SMOOTH THE DATA CURVES
      IT IS NOT NORMALLY USED IN ORDER TO PREVENT BIASING INPUT TIME
      SERIES DATA, BUT MAY BE USED AT USER'S DISCRETION. IF USED, NOTE
      THAT THE PROGRAM OUTPUT WILL NOT BE AS EXACT AS POSSIBLE

DO 73 L2=1,2
Q=0
DO 74 IS=1,65318
      SUMX=0.0
      SUMY=0.0
      SUMZ=0.0
DO 75 J=1,144
      SUMX=ZZX1(Q+J)+SUMX
      SUMY=ZZV1(Q+J)+SUMY
      SUMZ=ZZV1(Q+J)+SUMZ
75 CONTINUE
      ZZX1(IS)=SUMX/144.
      ZZV1(IS)=SUMY/144.
      Q=Q+1
74 CONTINUE
73 CONTINUE

```

```

APPI 0580
APPI 0590
APPI 0600
APPI 0610
APPI 0620
APPI 0630
APPI 0640
APPI 0650
APPI 0660
APPI 0670
APPI 0680
APPI 0690
APPI 0700
APPI 0710
APPI 0720
APPI 0730
APPI 0740
APPI 0750
APPI 0760
APPI 0770
APPI 0780
APPI 0790
APPI 0800
APPI 0810
APPI 0820
APPI 0830
APPI 0840
APPI 0850
APPI 0860
APPI 0870
APPI 0880
APPI 0890
APPI 0900
APPI 0910
APPI 0920
APPI 0930
APPI 0940
APPI 0950
APPI 0960
APPI 0970
APPI 0980
APPI 0990
APPI 1000
APPI 1010
APPI 1020
APPI 1030
APPI 1040
APPI 1050

```



```

CCCCCCCCCCCC C C C
REAL*8 AH(4),BH(4),CH(4),DH(4),EH(4),FH(4)
REAL*8 AL(3),BL(3),CL(3),DL(3),EL(3),FL(3)
DIMENSION FRQH(4),FRQL(3)

DEFINE AND COMPUTE ALL COEFFICIENTS

UNDER THE PRESENT DATA COLLECTION SYSTEM, 64 SAMPLES ARE TAKEN
PER SECOND. IF ANOTHER DATA COLLECTION SYSTEM IS USED, IT MUST
BE ADJUSTED TO THE SAMPLE PERIOD, I.E., 1/SAMPLE RATE

T=1./64.

COEFFICIENTS FOR FIXED HIGH PASS FILTER
AFHP1=-((T**2/160.-2.)/(1.+T/8.+T**2/320.))
AFHP2=-((1.-T/8.+T**2/320.)/(1.+T/8.+T**2/320.))
BFHP0=(1./(1.+T/8.+T**2/320.))
BFHP1=-((2.)/(1.+T/8.+T**2/320.))
BFHP2=(1./(1.+T/8.+T**2/320.))
WRITE(6,1002)AFHP1,AFHP2,BFHP0,BFHP1,BFHP2
FORMAT(1X,'FIXED FILTER AFHP1=',F19.16,'AFHP2=',F19.16,'BFHP0=',
$F19.16,'BFHP1=',F19.16,'BFHP2=',F19.16)
1002

COEFFICIENTS FOR SELECTABLE HIGH PASS FILTERS

THE FOLLOWING ARRAY VALUES ARE FIXED COEFFICIENTS FOR THE VARIOUS
FREQUENCY SELECTIONS POSSIBLE ON THE AN/ASQ-81
FRQH(1)=0.04
FRQH(2)=0.06
FRQH(3)=0.08
FRQH(4)=0.10
AH(1)=12.52096/40.82834
BH(1)=1./40.82834
CH(1)=11.00599/45.28317
DH(1)=1./45.28317
EH(1)=7.41498/57.57668
FH(1)=1./57.57668
AH(2)=8.34727/18.14591
BH(2)=1./18.14591
CH(2)=7.33959/20.12587
DH(2)=1./20.12587
EH(2)=4.94332/25.58964
FH(2)=1./25.58964
AH(3)=6.26045/10.20708
BH(3)=1./10.20708
APPL11060
APPL11070
APPL11080
APPL11090
APPL11100
APPL11110
APPL11120
APPL11130
APPL11140
APPL11150
APPL11160
APPL11170
APPL11180
APPL11190
APPL11200
APPL11210
APPL11220
APPL11230
APPL11240
APPL11250
APPL11260
APPL11270
APPL11280
APPL11290
APPL11300
APPL11310
APPL11320
APPL11330
APPL11340
APPL11350
APPL11360
APPL11370
APPL11380
APPL11390
APPL11400
APPL11410
APPL11420
APPL11430
APPL11440
APPL11450
APPL11460
APPL11470
APPL11480
APPL11490
APPL11500
APPL11510
APPL11520
APPL11530

```



```

CH(3)=5.50500/11.32080
DH(3)=1./11.32080
EH(3)=3.70749/14.39417
FH(3)=1./14.39417
AH(4)=5.00836/6.53253
BH(4)=1./6.53253
CH(4)=4.40400/7.24531
DH(4)=1./7.24531
EH(4)=2.96599/9.21227
FH(4)=1./9.21227

```

CCCCCCCC C

```

SELECT THE HIGH PASS FILTER SETTING
FOR THE LOW FREQUENCY CUTOFF AT 0.04 HZ, SET I=1
FOR THE LOW FREQUENCY CUTOFF AT 0.06 HZ, SET I=2
FOR THE LOW FREQUENCY CUTOFF AT 0.08 HZ, SET I=3
FOR THE LOW FREQUENCY CUTOFF AT 0.10 HZ, SET I=4

```

I=1

```

A1=1.+AH(I)*T/2.+BH(I)*(T**2)/4.
B1=-2.+BH(I)*(T**2/2.)
C1=1.-AH(I)*T/2.+BH(I)*(T**2)/4.
D1=1.+CH(I)*T/2.+DH(I)*(T**2)/4.
E1=-2.+DH(I)*(T**2)/2.
F1=1.-CH(I)*T/2.+DH(I)*(T**2)/4.
G1=1.+EH(I)*T/2.+FH(I)*(T**2)/4.
H1=-2.+FH(I)*(T**2)/2.
I1=1.-EH(I)*T/2.+FH(I)*(T**2)/4.

```

```

CODE IS "ASHP1" MEANS "A1 COEFFICIENT FOR THE SELECTABLE HIGH
PASS FILTER"

```

CCCC

```

ASHP1=1./(A1*D1*G1)
ASHP2=- (C1/A1)
ASHP3=- (B1/A1)
ASHP4=- (E1/D1)
ASHP5=- (F1/D1)
ASHP6=- (H1/G1)
ASHP7=- (I1/G1)

```

```

C1000 WRITE(6,1000)FRQH(I),ASHP1,ASHP2,ASHP3,ASHP4,ASHP5,ASHP6,ASHP7
FORMAT(IX,F5.3,ASHP1=,F19.16,ASHP2=,F19.16,ASHP3=,
,F19.16,ASHP4=,F19.16,ASHP5=,F19.16,ASHP6=,F19.16,
,ASHP7=,F19.16)

```

CCCCCCCC

```

COEFFICIENTS FOR SELECTABLE LOW PASS FILTERS

```

```

FRQL(1)=0.2
FRQL(2)=0.4

```

```

APPI11540
APPI11550
APPI11560
APPI11570
APPI11580
APPI11590
APPI11600
APPI11610
APPI11620
APPI11630
APPI11640
APPI11650
APPI11660
APPI11670
APPI11680
APPI11690
APPI11700
APPI11710
APPI11720
APPI11730
APPI11740
APPI11750
APPI11760
APPI11770
APPI11780
APPI11790
APPI11800
APPI11810
APPI11820
APPI11830
APPI11840
APPI11850
APPI11860
APPI11870
APPI11880
APPI11890
APPI11900
APPI11910
APPI11920
APPI11930
APPI11940
APPI11950
APPI11960
APPI11970
APPI11980
APPI11990
APPI12000
APPI12010

```



```

FRQL(3)=0.6
AL(3)=1./0. C3492
BL(3)=0.35804/0.03492
CL(3)=1./0. C3492
DL(3)=1./0.02779
EL(3)=0.20696/0.02779
FL(3)=1./0.02779
AL(1)=1./0.3143
BL(1)=1./0.3143
CL(1)=1./0.2501
DL(1)=1./0.2501
EL(1)=0.6205/0.2501
FL(1)=1./0.2501
AL(2)=1./0.07858
BL(2)=0.53706/0.07858
CL(2)=1./0.07858
DL(2)=1./0.06252
EL(2)=0.31044/0.06252
FL(2)=1./0.06252

```

```

SELECT LOW PASS FILTER SETTING AT 0.2 HZ, SET J=1
FOR THE HIGH FREQUENCY CUTOFF AT 0.4 HZ, SET J=2
FOR THE HIGH FREQUENCY CUTOFF AT 0.6 HZ, SET J=3

```

J=3

```

A1=AL(J)*(T**2)/4.
B1=2.*A1
C1=A1
D1=DL(J)*(T**2)/4.
E1=D1
F1=D1
G1=(1.-BL(J)*T/2.+CL(J)*(T**2)/4.)
H1=(-2.+CL(J)*(T**2)/2.)
I1=(1.-BL(J)*T/2.+CL(J)*(T**2)/4.)
J1=(1.+EL(J)*T/2.+FL(J)*(T**2)/4.)
K1=(-2.+FL(J)*(T**2)/2.)
L1=(1.-EL(J)*T/2.+FL(J)*(T**2)/4.)
ASLP1=- (G1*K1+H1*J1)/(G1*J1)
ASLP2=- (G1*L1+H1*K1+I1*J1)/(G1*J1)
ASLP3=- (H1*L1+I1*K1)/(G1*J1)
ASLP4=- (I1*L1)/(G1*J1)
BSLP0=(A1*D1)/(G1*J1)
BSLP1=(A1*E1+B1*D1)/(G1*J1)
BSLP2=(A1*F1+B1*E1+C1*D1)/(G1*J1)
BSLP3=(B1*F1+C1*E1)/(G1*J1)

```

CCCCCCC C

APPL2020
APPL2030
APPL2040
APPL2050
APPL2060
APPL2070
APPL2080
APPL2090
APPL2100
APPL2110
APPL2120
APPL2130
APPL2140
APPL2150
APPL2160
APPL2170
APPL2180
APPL2190
APPL2200
APPL2210
APPL2220
APPL2230
APPL2240
APPL2250
APPL2260
APPL2270
APPL2280
APPL2290
APPL2300
APPL2310
APPL2320
APPL2330
APPL2340
APPL2350
APPL2360
APPL2370
APPL2380
APPL2390
APPL2400
APPL2410
APPL2420
APPL2430
APPL2440
APPL2450
APPL2460
APPL2470
APPL2480
APPL2490


```

BSLP4=(C1*F1)/(G1*J1)
WRITE(6,1001)FRQL(J),ASLP1,ASLP2,ASLP3,ASLP4,BSLP0,BSLP1,BSLP2,
$BSLP3,BSLP4
C1001$FORMAI(IX,FREQ=F19.16,ASLP1=F19.16,ASLP2=F19.16,ASLP3=
$,F19.16,ASLP4=F19.16,BSLP0=F19.16,BSLP1=F19.16,
$,BSLP2=F19.16,BSLP3=F19.16,BSLP4=F19.16)
DO 100 I=1,24576
SIG=ZZXI(I)
Y0=BFHP0*SIG+BFHP1*SIG1+BFHP2*SIG2+AFHP1*Y01+AFHP2*Y02
XI=ASHP1*Y0+ASHP2*XI2+ASHP3*XI1
XI1=XI+XI2-2.*XI1
XI11=XI1+ASHP4*XI11+ASHP5*XI112
XI1V=XI11-2.*XI11+XI112
XV=XIV+ASHP6*XV1+ASHP7*XV2
YPO=XV+XV2-2.*XV1
GP1=ASLP1*OUTFD1+ASLP2*OUTFD2+ASLP3*OUTFD3+ASLP4*OUTFD4
GP2=BSLP0*YPO+BSLP1*YPO1+BSLP2*YPO2+BSLP3*YPO3+BSLP4*YPO4
OUTFLD(I)=GP1+GP2
FINISHED COMPUTING THIS STEP'S VALUES FOR AMPLITUDES
INCREMENT STORAGE REGISTERS
C C C C
SIG2=SIG1
SIG1=SIG
Y02=Y01
Y01=Y0
XI2=XI1
XI1=XI
XI112=XI111
XI111=XI11
XV2=XV1
XV1=XV
YPO4=YPO3
YPO3=YPO2
YPO2=YPO1
YPO1=YPO
OUTFD4=OUTFD3
OUTFD3=OUTFD2
OUTFD2=OUTFD1
OUTFD1=OUTFLD(I)
FINISHED INCREMENTING STORAGE REGISTERS
C C C
100 CONTINUE
XMAXP=TIME2(16384)
C C C
VERSATEC PLOT OF B - FIELD SPECTRA

```

```

APP12500
APP12510
APP12520
APP12530
APP12540
APP12550
APP12560
APP12570
APP12580
APP12590
APP12600
APP12610
APP12620
APP12630
APP12640
APP12650
APP12660
APP12670
APP12680
APP12690
APP12700
APP12710
APP12720
APP12730
APP12740
APP12750
APP12760
APP12770
APP12780
APP12790
APP12800
APP12810
APP12820
APP12830
APP12840
APP12850
APP12860
APP12870
APP12880
APP12890
APP12900
APP12910
APP12920
APP12930
APP12940
APP12950
APP12960
APP12970

```



```

C      NPTS=1020./CELTAT +1.
C      NPPTS=24576
C      NPPTS, NPTS DETERMINES NUMBER OF POINTS NECESSARY IN ORDER FOR
C      THE 0 TO 2041 SECS RANGE TO BE PLOTTED. REVIEW THE WRITE-UP
C      FOR THE FOLLOWING ITB, AND RTB VALUES.
C      FOR THE SUBROUTINE PROCEDURE 'DRAWP'.
      ITB(3)=8
      ITB(4)=4
      ITB(7)=1
      ITB(12)=0
      RTB(1)=0.0
      RTB(2)=0.0
      RTB(3)=ALAB(1)
      READ(5,3000)ITITLE
C      DRAW THE COIL ANTENNA TOTAL FIELD DATA SERIES
C      CALL DRAWP(NPTS,TIME2,ZZX1,ITB,RTB)
C      RTB(3)=ALAB(2)
C      READ(5,3000)ITITLE
C      DRAW THE ASQ81 TOTAL FIELD DATA SERIES
C      CALL DRAWP(NPTS,TIME2,ZZY1,ITB,RTB)
C      RTB(3)=ALAB(3)
C      READ(5,3000)ITITLE
C      DRAW THE SCHONSTEDT COIL FIELD DATA SERIES
C      CALL DRAWP(NPTS,TIME2,ZZV1,ITB,RTB)
C      RTB(3)=ALAB(4)
C      READ(5,3000)ITITLE
C      DRAW THE PROGRAM OUTPUT TOTAL FIELD DATA SERIES
C      CALL DRAWP(NPTS,TIME2,OUTFLD,ITB,RTB)
C      DRAW THE RAW COIL TIME SERIES DATA
C      RTB(3)=ALAB(4)
C      READ(5,3000)ITITLE
C      CALL DRAWP(NPTS,TIME2,CLFLD,ITB,RTB)
C      CONTINUE
C      FORMAT(6A8)
C      200 STOP
C      3000 END
C      SUBROUTINE RD(IUN,IO,IRS,IREC,IKQ)
C
APP12980
APP12990
APP13000
APP13010
APP13020
APP13030
APP13040
APP13050
APP13060
APP13070
APP13080
APP13090
APP13100
APP13110
APP13120
APP13130
APP13140
APP13150
APP13160
APP13170
APP13180
APP13190
APP13200
APP13210
APP13220
APP13230
APP13240
APP13250
APP13260
APP13270
APP13280
APP13290
APP13300
APP13310
APP13320
APP13330
APP13340
APP13350
APP13360
APP13370
APP13380
APP13390
APP13400
APP13410
APP13420
APP13430
APP13440
APP13450

```



```

THIS PROCEDURE FURNISHED BY DR. TIM STANTON,
DEPARTMENT OF OCEANOGRAPHY.

      READ DATA FROM IUN, ALIGN , CHECK & RETURN

IUN=TAPE NUMBER, EG 20
IQ=INTEGER*2 ARRAY, 16 LONG, (VALUES 0-4095, SUBTRACT 2048)*5
/2028. GIVES VOLTAGE
IRS= NUMBER OF RESINCS ALLOWED (ERRORS)
IREC= COUNTER OF RECORDS (FRAMES OF DATA)
      BLOCK 512 BITS, 32 BITS = RECORD
      800 BPI TAPE UNLABELED
      IRQ= NUMBER OF ACTUAL RESINCS (ERRORS)

      INTEGER * 2 IO(16),IP(16)
DATA IRR /0/
      IF ( IREC.EQ.0 ) IS=0
      IERR=0
      FORMAT (16A2)
      IF ( IS.NE.0 ) GO TO 50
      READ ( IUN,20,END=900) IP
      IREC=IREC+1
      IS=IS+1
      IF ( IS.LT.17) GO TO 50
      READ ( IUN,20,END=900) IP
      IS=1
      IREC=IREC+1
      ICH=IMASK(IP(IS),3,0)+1
      WRITE (6,55) ICH,IS,IUN,IREC
      FORMAT ( , RESYNCRNG ICH,IS,IUN,IREC ,4I8)

      IF ( ICH.NE.1) GO TO 40
      DO 100 I=1,16
      IO(I)=ISHIFT(IP(IS),4)
      ICH=IMASK(IP(IS),3,0)+1
      IF ( ICH.EQ.1) GO TO 80
      IERR=IER+1
      WRITE (6,70) IUN,IREC ,I,ICH,IERR
      FORMAT ( , IUN , ,13, , RECORD , ,16, ,CHAN & DATA CH ,2I4,
      , ERRORS , ,I7)
      IS=IS+1
      IF ( IS.LT.17) GO TO 100
      READ ( IUN,20,END=900) IP
      IS=1
      IREC=IREC+1
      CONTINUE
100
C

```

CCCCCCCCCCCCCCCC

20

40

50

C 55

C

70

80

100

C


```

110 IF ( IER.EQ.0) GO TO 150
    IRR=IRR+1
    IF (IRR.LT.IRS) GO TO 120
    WRITE (6,110)
    FORMAT (11 STOPPED IN SUB RD BECAUSE OF IRR.GT.,I6,' AT L110')
    IRR=IRR
    STOP
120 CONTINUE
    WRITE (6,130) IREC,IRR
    FORMAT (1 RESYNC AT FRAME ',I6,' WITH TOTAL ERRORS ',I7)
    IER=0
    IRR=IRR
    GO TO 50
150 CONTINUE
    RETURN
900 WRITE (6,910) IUN,IREC
910 FORMAT (11 END OF UNIT ',I3,' AT REC ',I7)
    STOP
    END

FUNCTION ISHIFT (IN,NPLC)
    RETURNS SHIFTED VALUE OF I*2 WORD IN
    -VE LEFT,+VE RIGHT SHIFT

    INTEGER * 2 IN
    IP=IN
    IF (IP.LT.0) IP=IP+65536
    IF (NPLC.LT.0) GO TO 30
    ISHIFT=IP/(2*IABS(NPLC))
    RETURN
    ISHIFT=IP*(2*IABS(NPLC))
    IF (ISHIFT.GT.65535) ISHIFT=MOD(ISHIFT,65536)
    RETURN
END
FUNCTION IMASK (IN,IBL,IBR)
    MASK I*2 WORD IN OUTSIDE BITS IBL & IBR

    INTEGER * 2 IN,I0
    I0=IN
    IF (IBR.EQ.0) GO TO 50
    IT=ISHIFT(IN,IBR)
    I0=IT
    IP=ISHIFT(I0,IBL-15-IBR)
    I0=IP
    IMASK=ISHIFT(I0,15-IBL)
    RETURN
END
SUBROUTINE FOUR

```

```

APP13940
APP13950
APP13960
APP13970
APP13980
APP13990
APP14000
APP14010
APP14020
APP14030
APP14040
APP14050
APP14060
APP14070
APP14080
APP14090
APP14100
APP14110
APP14120
APP14130
APP14140
APP14150
APP14160
APP14170
APP14180
APP14190
APP14200
APP14210
APP14220
APP14230
APP14240
APP14250
APP14260
APP14270
APP14280
APP14290
APP14300
APP14310
APP14320
APP14330
APP14340
APP14350
APP14360
APP14370
APP14380
APP14390
APP14400
APP14410

```


APPI 4420
 APPI 4430
 APPI 4440
 APPI 4450
 APPI 4460
 APPI 4470
 APPI 4480
 APPI 4490
 APPI 4500
 APPI 4510
 APPI 4520
 APPI 4530
 APPI 4540
 APPI 4550
 APPI 4560
 APPI 4570
 APPI 4580
 APPI 4590
 APPI 4600
 APPI 4610
 APPI 4620
 APPI 4630
 APPI 4640
 APPI 4650
 APPI 4660
 APPI 4670
 APPI 4680
 APPI 4690
 APPI 4700
 APPI 4710
 APPI 4720
 APPI 4730
 APPI 4740
 APPI 4750
 APPI 4760
 APPI 4770
 APPI 4780
 APPI 4790
 APPI 4800
 APPI 4810
 APPI 4820
 APPI 4830
 APPI 4840
 APPI 4850
 APPI 4860
 APPI 4870
 APPI 4880
 APPI 4890

PURPOSE

SUBROUTINE FOURT COMPUTES THE FORWARD AND INVERSE COOLEY-TUKEY FAST FOURIER TRANSFORM OF THE CONTENTS OF THE ARRAY DATA. FOR DATA A SINGLY-DIMENSIONED ARRAY OF LENGTH L, THE JTH COMPONENT OF THE TRANSFORM IS GIVEN BY $SUM(DATA(K)*W**(K-1)*(J-1))$ WHERE THE SUM IS TAKEN OVER K, 1 .LE. K .LE. L, AND $W=EXP(ISIGN*2*PI*SQRT(-1)/L)$

THE VALUE OF ISIGN DEPENDS UPON WHETHER A FORWARD OR INVERSE TRANSFORM IS TO BE PERFORMED. FOURT MAY ALSO BE USED ON A MULTI-DIMENSIONAL ARRAY, IN WHICH CASE A FOURIER TRANSFORM IS PERFORMED ALONG EACH DIMENSION IN TURN.

CALLING SEQUENCE

CALL FOURT(DATA,NN,NDIM,ISIGN,IFORM,WORK)

DESCRIPTION OF ARGUMENTS

DATA COMPLEX*8 MULTI-DIMENSIONAL ARRAY CONTAINING THE DATA TO BE TRANSFORMED. ON OUTPUT DATA CONTAINS THE TRANSFORM. NORMAL FORTRAN ORDERING IS EXPECTED, THE FIRST SUBSCRIPT CHANGING THE FASTEST.

NN INTEGER*4 ARRAY CONTAINING THE DIMENSIONS OF THE ARRAY DATA.

NDIM NUMBER OF DIMENSIONS OF THE ARRAY DATA = NUMBER OF ELEMENTS IN THE ARRAY NN.

ISIGN INTEGER INDICATING WHETHER FORWARD OR INVERSE TRANSFORM IS TO BE PERFORMED.
 ISIGN=-1 FOR FORWARD TRANSFORM
 ISIGN=1 FOR INVERSE TRANSFORM.
 NOTE: THESE DEFINITIONS ARE NOT STANDARDIZED. IN PARTICULAR, THE DEFINITIONS OF FORWARD AND INVERSE TRANSFORM ARE REVERSED IN THE IMSL FFT ROUTINES.

IFORM AN INTEGER INDICATING WHETHER OR NOT DATA CONTAINS ONLY PURELY REAL VALUES.
 IFORM=0 IF DATA IS PURELY REAL
 IFORM=1 OTHERWISE.
 IF IFORM IS SET TO 0, ALL THE IMAGINARY PARTS OF THE ELEMENTS IN DATA MUST BE SET TO 0.0.

WORK A 1-DIMENSIONAL REAL*4 ARRAY USED FOR WORKING STORAGE.
 ITS LENGTH SHOULD BE TWICE THE LARGEST ARRAY DIMENSION
 NN(1), I=1,2,...,NDIM, WHICH IS NOT A POWER OF TWO. IN
 PARTICULAR, IF ALL NN(I) ARE POWERS OF TWO, NO WORK SPACE
 IS NEEDED AND WORK MAY BE REPLACED BY ZERO IN THE CALLING
 SEQUENCE.

REMARKS

IF AN INVERSE TRANSFORM (ISIGN=+1) IS PERFORMED UPON AN ARRAY
 OF TRANSFORMED (ISIGN=-1) DATA, THE ORIGINAL DATA WILL REAP-
 PEAR, MULTIPLIED BY NN(1)*NN(2)*...*NN(NDIM).

FOR A MULTI-DIMENSIONAL ARRAY THE (J1,J2,...,JNDIM)
 COMPONENT OF THE TRANSFORM IS GIVEN BY

SUM(DATA(I1,I2,...,INDIM)*W1**((I1-1)*(J1-1))*

W2**((I2-1)*(J2-1))*...*WNDIM**((INDIM-1)*(JNDIM-1)))

HERE THE SUM RANGES OVER ALL POSSIBLE VALUES OF THE I'S

AND W1=EXP(ISIGN*2*PI*SQRT((-1)/NN(1)), ETC.

THE ARRAY OF INPUT DATA MUST BE IN COMPLEX FORMAT.
 HOWEVER, IS ALL IMAGINARY PARTS ARE ZERO (I.E., THE DATA
 ARE DISGUISED REAL) RUNNING TIME IS CUT UP TO FORTY PER-
 CENT. (FOR FASTEST TRANSFORM OF REAL DATA, NN(1) SHOULD BE E-
 VEN.) THE TRANSFORM VALUES ARE ALWAYS COMPLEX AND ARE RETURNED
 IN THE ORIGINAL ARRAY OF DATA, REPLACING THE INPUT DATA. THE
 LENGTH OF EACH DIMENSION OF THE DATA ARRAY MAY BE ANY INTEGER.
 THE PROGRAM RUNS FASTER ON COMPOSITE INTEGERS THAN ON PRIMES,
 AND IS PARTICULARLY FAST ON NUMBERS RICH IN FACTORS OF TWO.

TIMING IS IN FACT GIVEN BY THE FOLLOWING FORMULA. LET NTOT BE
 THE TOTAL NUMBER OF POINTS (REAL OR COMPLEX) IN THE DATA ARRAY,
 THAT IS, NTOT=NN(1)*NN(2)*... DECOMPOSE NTOT INTO ITS PRIME
 FACTORS, SUCH AS 2**K2*3**K3*5**K5*... LET SUM2 BE THE
 SUM OF THE FACTORS OF TWO IN NTOT, THAT IS, SUM2 = 2*K2. LET
 SUMF BE THE SUM OF ALL OTHER FACTORS OF NTOT, THAT IS, SUMF =
 3*K3*5*K5*... THE TIME TAKEN BY A MULTI-DIMENSIONAL TRANSFORM ON
 THESE NTOT DATA IS TIME TO + NTOT*(T1+T2*SUM2+T3*SUMF). ON THE
 CDC 3300 (FLOATING POINT ADD TIME = SIX MICROSECONDS), T = 3000+
 NTOT*(600+40*SUM2+175*SUMF) MICROSECONDS ON COMPLEX DATA.

THE SAVINGS OFFERED BY THIS PROGRAM CAN BE DRAMATIC. A ONE-DI-
 MENSIONAL ARRAY 4000 IN LENGTH WILL BE TRANSFORMED IN 4000*(600+
 40*(2+2+2+2)+175*(5+5+5)) = 14.5 SECONDS VERSUS ABOUT 4000*
 4000*175 = 2800 SECONDS FOR THE STRAIGHTFORWARD TECHNIQUE.

THE FAST FOURIER TRANSFORM PLACES THREE RESTRICTIONS UPON THE
 DATA.

APPI 4900
 APPI 4910
 APPI 4920
 APPI 4930
 APPI 4940
 APPI 4950
 APPI 4960
 APPI 4970
 APPI 4980
 APPI 4990
 APPI 5000
 APPI 5010
 APPI 5020
 APPI 5030
 APPI 5040
 APPI 5050
 APPI 5060
 APPI 5070
 APPI 5080
 APPI 5090
 APPI 5100
 APPI 5110
 APPI 5120
 APPI 5130
 APPI 5140
 APPI 5150
 APPI 5160
 APPI 5170
 APPI 5180
 APPI 5190
 APPI 5200
 APPI 5210
 APPI 5220
 APPI 5230
 APPI 5240
 APPI 5250
 APPI 5260
 APPI 5270
 APPI 5280
 APPI 5290
 APPI 5300
 APPI 5310
 APPI 5320
 APPI 5330
 APPI 5340
 APPI 5350
 APPI 5360
 APPI 5370

1. THE NUMBER OF INPUT DATA AND THE NUMBER OF TRANSFORM VALUES MUST BE THE SAME.
2. BOTH THE INPUT DATA AND THE TRANSFORM VALUES MUST REPRESENT EQUISPACED POINTS IN THEIR RESPECTIVE DOMAINS OF TIME AND FREQUENCY. CALLING THESE SPACINGS DELTAT AND DELTAF, IT MUST BE TRUE THAT $\text{DELTAF} = 2 * \pi / (\text{NN}(I) * \text{DELTAT})$. OF COURSE, DELTAT NEED NOT BE THE SAME FOR EVERY DIMENSION.
3. CONCEPTUALLY AT LEAST, THE INPUT DATA AND THE TRANSFORM OUTPUT REPRESENT SINGLE CYCLES OF PERIODIC FUNCTIONS.

THERE ARE NO ERROR MESSAGES OR ERROR HALTS IN THIS PROGRAM. THE PROGRAM RETURNS IMMEDIATELY IF NDM OR ANY $\text{NN}(I)$ IS LESS THAN ONE.

FOR MOST APPLICATIONS FOURT, IF COMPILED UNDER FORTRAN H, IS COMPARABLE IN SPEED AND ACCURACY TO THE IMSL FFT SUBROUTINES. WITH CERTAIN PATHOLOGICALLY ILL-CONDITIONED DATA THE ACCURACY OF FOURT MAY BE SERIOUSLY DEGRADED, BUT THE SAME CAN PROBABLY BE SAID OF ANY EXTANT FFT ROUTINE. WORK SPACE REQUIRED BY FOURT MAY BE GREATER OR LESS THAN THAT REQUIRED BY THE IMSL ROUTINES, DEPENDING UPON THE APPLICATION. FOURT IS MORE FLEXIBLE AND IN GENERAL EASIER TO USE THAN THE IMSL ROUTINES. FOURT ALONE PROVIDES THE CAPABILITY OF TRANSFORMING A MULTI-DIMENSIONAL ARRAY WITH A SINGLE CALL.

THIS IS THE FASTEST AND MOST VERSATILE VERSION OF THE FFT KNOWN TO THE AUTHOR. A PROGRAM CALLED FOUR2 IS AVAILABLE THAT ALSO PERFORMS THE FAST FOURIER TRANSFORM AND IS WRITTEN IN USASI BASIC FORTRAN. IT IS ABOUT ONE THIRD AS LONG AND RESTRICTS THE DIMENSIONS OF THE INPUT ARRAY (WHICH MUST BE COMPLEX) TO BE POWERS OF TWO. ANOTHER PROGRAM, CALLED FOUR1, IS ONE TENTH AS LONG AND RUNS TWO THIRDS AS FAST ON A ONE-DIMENSIONAL COMPLEX ARRAY WHOSE LENGTH IS A POWER OF TWO.

REFERENCE--

IEEE AUDIO TRANSACTIONS (JUNE 1967), SPECIAL ISSUE ON THE FFT.

EXAMPLE 1. THREE-DIMENSIONAL FORWARD FOURIER TRANSFORM OF A COMPLEX ARRAY DIMENSIONED 32 BY 25 BY 13 IN FORTRAN IV.

```

DIMENSION DATA(32,25,13),WORK(50),NN(3)
COMPLEX DATA
DATA NN/32,25,13/
DO 1 I=1,32
DO 1 J=1,25
DO 1 K=1,13
DATA(I,J,K)=COMPLEX VALUE
CALL FOURT(DATA,NN,3,-1,1,WORK)

```

APPL5380
APPL5390
APPL5400
APPL5410
APPL5420
APPL5430
APPL5440
APPL5450
APPL5460
APPL5470
APPL5480
APPL5490
APPL5500
APPL5510
APPL5520
APPL5530
APPL5540
APPL5550
APPL5560
APPL5570
APPL5580
APPL5590
APPL5600
APPL5610
APPL5620
APPL5630
APPL5640
APPL5650
APPL5660
APPL5670
APPL5680
APPL5690
APPL5700
APPL5710
APPL5720
APPL5730
APPL5740
APPL5750
APPL5760
APPL5770
APPL5780
APPL5790
APPL5800
APPL5810
APPL5820
APPL5830
APPL5840
APPL5850


```

      I=1      80      J=1,NTOT
      DO 80 J=1,NTOT
      DATA(J)=DATA(I)
      I=I+2
C
C
C
C
      SHUFFLE DATA BY BIT REVERSAL, SINCE N=2**K. AS THE SHUFFLING
      CAN BE DONE BY SIMPLE INTERCHANGE, NO WORKING ARRAY IS NEEDED
C
      100      IF (NTWO-NP2)200,110,110
      110      NP2HF=NP2/2
      J=1
      DO 150 I2=1,NP2,NP1
      IF (J-I2)120,130,130
      120      I1MAX=I2+NP1-2
      DO 125 I1=I2,I1MAX,2
      DO 125 I3=I1,NTOT,NP2
      J3=J+I3-I2
      TEMPR=DATA(I3)
      TEMPI=DATA(I3+1)
      DATA(I3)=DATA(J3)
      DATA(I3+1)=DATA(J3+1)
      DATA(J3)=TEMPR
      DATA(J3+1)=TEMPI
      M=NP2HF
      125      IF (J-M)150,150,145
      130      J=J-M
      140      M=M/2
      145      J=J+M
      150      IF (M-NP1)150,140,140
      J=J+M
      GO TO 300
C
C
C
      SHUFFLE DATA BY DIGIT REVERSAL FOR GENERAL NG ARRAY IS NEEDED
C
      200      NWORKE=2*N
      DO 270 I1=1,NP1,2
      DO 270 I3=I1,NTOT,NP2
      J=I3
      DO 260 I=1,NWORK,2
      IF (ICASE-3)210,220,210
      210      WORK(I)=DATA(J)
      WORK(I+1)=DATA(J+1)
      GO TO 230
      220      WORK(I)=DATA(J)
      WORK(I+1)=0.
      230      IFP2=NP2
      IF=IFMIN
      240      IFP1=IFP2/IFACT(IF)
      J=J+IFP1

```

```

APPI6820
APPI6830
APPI6840
APPI6850
APPI6860
APPI6870
APPI6880
APPI6890
APPI6900
APPI6910
APPI6920
APPI6930
APPI6940
APPI6950
APPI6960
APPI6970
APPI6980
APPI6990
APPI7000
APPI7010
APPI7020
APPI7030
APPI7040
APPI7050
APPI7060
APPI7070
APPI7080
APPI7090
APPI7100
APPI7110
APPI7120
APPI7130
APPI7140
APPI7150
APPI7160
APPI7170
APPI7180
APPI7190
APPI7200
APPI7210
APPI7220
APPI7230
APPI7240
APPI7250
APPI7260
APPI7270
APPI7280
APPI7290

```



```

250 IF(J-I3-IFP2)260,250,250
    J=J-IFP2
    IFP2=IFP1
260 IF=IF+1
    IF(IFP2-NP1)260,260,240
    CONTINUE
    I2MAX=I3+NP2-NP1
    I=1
    DO 270 I2=I3,I2MAX,NP1
        DATA(I2)=WORK(I)
        DATA(I2+1)=WORK(I+1)
        I=I+2
270 C
C
C
C
C
C
    MAIN LOOP FOR FACTORS OF TWO. PERFORM FOURIER TRANSFORMS OF
    LENGTH FOUR, WITH ONE OF LENGTH TWO IF NEEDED. THE TWIDDLE FAC-
    TOR W=EXP(I*SIGN*2*PI*SQRT((-1)**M/(4*M*MAX))). CHECK FOR W=ISIGN*
    SQRT(-1)
300 IF (NTWO-NP1)600,600,305
305 NP1TW=NP1+NP1
    IPAR=NTWO/NP1
310 IF (IPAR-2)350,330,320
320 IPAR=IPAR/4
    GO TO 310
330 DO 340 I1=1,I1RNG,2
    DO 340 K1=1,1,NTOT,NP1TW
        K2=K1+NP1
        TEMPR=DATA(K2)
        TEMPI=DATA(K2+1)
        DATA(K2)=DATA(K1)-TEMPR
        DATA(K2+1)=DATA(K1+1)-TEMPI
        DATA(K1)=DATA(K1)+TEMPR
        DATA(K1+1)=DATA(K1+1)+TEMPI
340 MMAX=NP1
350 IF (MMAX-NTWO/2)370,600,600
360 LMAX=MAX0(NP1TW,MMAX/2)
370 DO 570 L=NP1,LMAX,NP1TW
    M=L
    IF (MMAX-NP1)420,420,380
    THETA=-TWOPI*FLOAT(L)/FLOAT(4*M*MAX)
380 IF (ISIGN)400,390,390
    THETA=-THETA
390 WR=COS(THETA)
400 WI=SIN(THETA)
    W2R=WR*WR-WI*WI
    W2I=2.*WR*WI
    W3R=W2R*WR-W2I*W2I
    W3I=W2R*WI+W2I*WR
410

```

```

APPL 7300
APPL 7310
APPL 7320
APPL 7330
APPL 7340
APPL 7350
APPL 7360
APPL 7370
APPL 7380
APPL 7390
APPL 7400
APPL 7410
APPL 7420
APPL 7430
APPL 7440
APPL 7450
APPL 7460
APPL 7470
APPL 7480
APPL 7490
APPL 7500
APPL 7510
APPL 7520
APPL 7530
APPL 7540
APPL 7550
APPL 7560
APPL 7570
APPL 7580
APPL 7590
APPL 7600
APPL 7610
APPL 7620
APPL 7630
APPL 7640
APPL 7650
APPL 7660
APPL 7670
APPL 7680
APPL 7690
APPL 7700
APPL 7710
APPL 7720
APPL 7730
APPL 7740
APPL 7750
APPL 7760
APPL 7770

```



```

420      DO 530 I1=1,IIRNG,2
      KM IN=I1+IPAR*M
      IF (MMAX-NP1)430,430,440
430      KM IN=I1
      KD IF=IPAR*MMAX
440      KSTEP=4*KD IF
450      IF (KSTEP-NTWO)460,460,530
460      DO 520 K1=K MIN,NTOT,KSTEP
      K2=K1+KD IF
      K3=K2+KD IF
      K4=K3+KD IF
      IF (MMAX-NP1)470,470,480
470      U1I=DATA(K1)+DATA(K2)
      U2I=DATA(K1+1)+DATA(K2+1)
      U2R=DATA(K3)+DATA(K4)
      U2I=DATA(K3+1)+DATA(K4+1)
      U3I=DATA(K1)-DATA(K2)
      U3R=DATA(K1+1)-DATA(K2+1)
      IF (ISIGN)471,472,472
471      U4R=DATA(K3+1)-DATA(K4+1)
      U4I=DATA(K4)-DATA(K3)
      GO TO 510
      U4R=DATA(K4+1)-DATA(K3+1)
472      U4I=DATA(K3)-DATA(K4)
      GO TO 510
480      T2R=W2R*DATA(K2)-W2I*DATA(K2+1)
      T2I=W2R*DATA(K2+1)+W2I*DATA(K2)
      T3R=W3R*DATA(K3)-W3I*DATA(K3+1)
      T3I=W3R*DATA(K3+1)+W3I*DATA(K3)
      T4R=W3R*DATA(K4)-W3I*DATA(K4+1)
      T4I=W3R*DATA(K4+1)+W3I*DATA(K4)
      U1I=DATA(K1)+T2R
      U1I=DATA(K1+1)+T2I
      U2R=T3R+T4R
      U2I=T3I+T4I
      U3I=DATA(K1+1)-T2I
      U3R=DATA(K1)-T2R
      IF (ISIGN)490,500,500
490      U4R=T3I-T4I
      U4I=T4R-T3R
      GO TO 510
      U4R=T4I-T3I
      U4I=T3R-T4R
500      DATA(K1)=U1R+U2R
      DATA(K1+1)=U1I+U2I
510      DATA(K2)=U3R+U4R
      DATA(K2+1)=U3I+U4I
      DATA(K3)=U1R-U2R

```

```

APP17780
APP17790
APP17800
APP17810
APP17820
APP17830
APP17840
APP17850
APP17860
APP17870
APP17880
APP17890
APP17900
APP17910
APP17920
APP17930
APP17940
APP17950
APP17960
APP17970
APP17980
APP17990
APP18000
APP18010
APP18020
APP18030
APP18040
APP18050
APP18060
APP18070
APP18080
APP18090
APP18100
APP18110
APP18120
APP18130
APP18140
APP18150
APP18160
APP18170
APP18180
APP18190
APP18200
APP18210
APP18220
APP18230
APP18240
APP18250

```



```

520      DATA(K3+1)=U1I-U2I
      DATA(K4)=U3R-U4R
      DATA(K4+1)=U3I-U4I
      KDIF=KSTEP
      KMIN=4*(KMIN-I1)+I1
      GO TO 450
530      CONTINUE
      M=M+LMAX
      IF (M-MMAX) 540,540,570
540      IF (ISIGN) 550,560,560
550      TEMPR=WR
      WR=(WR+W1)*RTHLF
      W1=(W1-TEMPR)*RTHLF
      GO TO 410
560      TEMPR=WR
      WR=(WR-W1)*RTHLF
      W1=(TEMPR+W1)*RTHLF
      GO TO 410
570      CONTINUE
      IPAR=3-IPAR
      MMAX=MMAX+MMAX
      GO TO 360

C      MAIN LOOP FOR FACTORS NOT EQUAL TO TWO. APPLY THE TWIDDLE FAC-
C      TOR W=EXP(I SIGN*2*PI*SQR((-1)*(J1-1))*(J2-J1)/(IFP1+IFP2));
C      THEN PERFORM A FOURIER TRANSFORM OF LENGTH IFACT(IF), MAKING USE
C      OF CONJUGATE SYMMETRIES.
      IF (NTWO-NP2) 605,700,700
      IFP1=NTWO
      IF=INON2
      NP1HF=NP1/2
      IFP2=IFACT(IF)*IFP1
      J1MIN=NP1+1
      IF (J1MIN-IFP1) 615,615,640
      DO 635 J1=J1MIN,IFP1,NP1
      THETA=-TWOP1*FLOAT(J1-1)/FLOAT(IFP2)
      IF (ISIGN) 625,620,620
      THETA=-THETA
      WSTPR=COS(THETA)
      WSTPI=SIN(THETA)
      WR=WSTPR
      W1=WSTPI
      J2MIN=J1+IFP1
      J2MAX=J1+IFP2-IFP1
      DO 635 J2=J2MIN,J2MAX,IFP1
      I1MAX=J2+I1RNG-2
      DO 630 I1=J2,I1MAX,2

```

```

APPI 8260
APPI 8270
APPI 8280
APPI 8290
APPI 8300
APPI 8310
APPI 8320
APPI 8330
APPI 8340
APPI 8350
APPI 8360
APPI 8370
APPI 8380
APPI 8390
APPI 8400
APPI 8410
APPI 8420
APPI 8430
APPI 8440
APPI 8450
APPI 8460
APPI 8470
APPI 8480
APPI 8490
APPI 8500
APPI 8510
APPI 8520
APPI 8530
APPI 8540
APPI 8550
APPI 8560
APPI 8570
APPI 8580
APPI 8590
APPI 8600
APPI 8610
APPI 8620
APPI 8630
APPI 8640
APPI 8650
APPI 8660
APPI 8670
APPI 8680
APPI 8690
APPI 8700
APPI 8710
APPI 8720
APPI 8730

```



```

630 J3=I1,NTOT,IFP2
TEMPR=DATA(J3)
DATA(J3)=DATA(J3)*WR-DATA(J3+1)*WI
DATA(J3+1)=TEMPR*WI+DATA(J3+1)*WR
TEMPR=WR
WR=WR*WSTPR-WI*WSTPI
WI=TEMPR*WSTPI+WI*WSTPR
THETA=-TWOPI/FOAT(IFACT(IF))
IF(ISIGN)650,645,645
THETA=-THETA
WSTPR=COS(THETA)
WSTPI=SIN(THETA)
J2RNG=IFP1*(1+IFACT(IF)/2)
DO 695 I1=1,I1RNG,2
DO 695 I3=1,NTOT,NP2
J2MAX=I3+J2RNG-IFP1
DO 690 J2=I3,J2MAX,IFP1
J1MAX=J2+IFP1-NP1
DO 680 J1=J2,J1MAX,NP1
J3MAX=J1+NP2-IFP2
DO 680 J3=J1,J3MAX,IFP2
JMIN=J3-J2+I3
JMAX=JMIN+IFP2-IFP1
I=1+(J3-I3)/NP1HF
IF(J2-I3)655,655,665
SUMR=0.
SUMI=0.
DO 660 J=JMIN,JMAX,IFP1
SUMR=SUMR+DATA(J)
SUMI=SUMI+DATA(J+1)
WORK(I)=SUMR
WORK(I+1)=SUMI
GO TO 680
ICONJ=1+(IFP2-2*J2+I3+J3)/NP1HF
J=JMAX
SUMR=DATA(J)
SUMI=DATA(J+1)
OLDSR=0.
OLDSI=0.
J=J-IFPI
TEMPR=SUMR
SUMR=TWOWR*SUMR-OLDSR+DATA(J)
TEMPI=SUMI
SUMI=TWOWR*SUMI-OLDSI+DATA(J+1)
OLDSR=TEMPR
OLDSI=TEMPI
J=J-IFPI
IF(J-JMIN)675,675,670

```

```

APP18740
APP18750
APP18760
APP18770
APP18780
APP18790
APP18800
APP18810
APP18820
APP18830
APP18840
APP18850
APP18860
APP18870
APP18880
APP18890
APP18900
APP18910
APP18920
APP18930
APP18940
APP18950
APP18960
APP18970
APP18980
APP18990
APP19000
APP19010
APP19020
APP19030
APP19040
APP19050
APP19060
APP19070
APP19080
APP19090
APP19100
APP19110
APP19120
APP19130
APP19140
APP19150
APP19160
APP19170
APP19180
APP19190
APP19200
APP19210

```



```

805      DO 860 I3=1,NTOT,NP2
      I2MAX=I3+NP2-NP1
      DO 860 I2=I3,I2MAX,NP1
      IMIN=I2+1,IRNG
      IMAX=I2+NP1-2
      JMAX=2*I3+NP1-IMIN
      IF(I2-I3)820,820,810
810      JMAX=JMAX+NP2
820      IF(IDIM-2)850,850,830
830      J=JMAX*NP0
      DO 840 I=IMIN,IMAX,2
      DATA(I)=DATA(J)
      DATA(I+1)=-DATA(J+1)
      J=J-2
840      J=JMAX
850      DO 860 I=IMIN,IMAX,NP0
      DATA(I)=DATA(J)
      DATA(I+1)=-DATA(J+1)
      J=J-NP0
860      END OF LOOP ON EACH DIMENSION
C
C
C
900      NP0=NP1
910      NP1=NP2
920      NPREV=N
      RETURN
      END
/*GO,SYSIN DD *
LA MESA VILLAGE, 13 MAY 83
COIL ANTENNA AMP, IN NT
LA MESA VILLAGE, 13 MAY 83
ASQ-81 AMP IN NT
LA MESA VILLAGE, 13 MAY 83
FLUXGATE AMP IN NT
LA MESA VILLAGE, 13 MAY 83
PROGRAM OUTPUT IN NT
LA MESA VILLAGE, 13 MAY 83
RAW COIL TIME SERIES IN VOLTS
LA MESA VILLAGE, 13 MAY 83
COIL ANTENNA AMP, IN NT
LA MESA VILLAGE, 13 MAY 83
ASQ81 AMP IN NT
LA MESA VILLAGE, 13 MAY 83
FLUXGATE AMP IN NT
LA MESA VILLAGE, 13 MAY 83
PROGRAM OUTPUT IN NT
LA MESA VILLAGE, 13 MAY 83

```

```

APP20180
APP20190
APP20200
APP20210
APP20220
APP20230
APP20240
APP20250
APP20260
APP20270
APP20280
APP20290
APP20300
APP20310
APP20320
APP20330
APP20340
APP20350
APP20360
APP20370
APP20380
APP20390
APP20400
APP20410
APP20420
APP20430
APP20440
APP20450
APP20460
APP20470
APP20480
APP20490
APP20500
APP20510
APP20520
APP20530
APP20540
APP20550
APP20560
APP20570
APP20580
APP20590
APP20600
APP20610
APP20620
APP20630
APP20640
APP20650

```



```

RAW COIL TIME SERIES IN VOLTS
/*
//GO.FT20F001 DD UNIT=3400-4,VOL=SER=MIKE1,DISP=(OLD,KEEP),
//      LABEL=(1,NL,IN),
//      DCB=(RECFM=FB,LRECL=32,BLKSIZE=512,DEN=2)
//GO.SYSDUMP DD SYSOUT=A,OUTLIM=65000

```

```

APP20660
APP20670
APP20680
APP20690
APP20700
APP20710

```


LIST OF REFERENCES

1. Air Force Geophysics Laboratory, L. G. Hanscom Field, Bedford, Massachusetts, Report AFGL-79-0192, Proceedings of the Air Force Geophysics Laboratory Workshop on Geomagnetism: April 6-7, 1979, Edited by R. C. Sagalyn, R. O. Hutchinson, and S. Gussenhoven, p. 47, 1979.
2. Jacobs, J. A., Geomagnetic Micropulsations, Springer-Verlag, New York, 1970.
3. Air Force Cambridge Research Laboratories, L. G. Hanscom Field, Bedford, Massachusetts, Report AFCRL-72-0570, The Geomagnetic Field, by David J. Knecht, 1972.
4. Naval Oceanographic Office, Washington, D. C., Technical Report TR-218, The Influence of the Natural Environment on MAD Operations, by James A. Brennan and Thomas M. Davis, September 1969.
5. Chilton, F., Wood, L., and Buntzen, R., "Electric and Magnetic Sensing Systems: Applications," chapter 10 of Applications of Remote Sensing to Ocean Surveillance, AGARD Lecture Series No. 88, North Atlantic Treaty Organization, Advisory Group for Aerospace Research and Development, London, 1977.
6. Naval Air Systems Command NAVAIR 16-30ASQ81-500, Preliminary Manual Interim Field Engineering Maintenance Instructions with Illustrated Parts Breakdown, Magnetic Detecting Set AN/ASQ-81(V)2, 1 November 1971.
7. Oppenheim, Alan V., and Schafer, Ronald W., Digital Signal Processing, Prentice-Hall, Inc., New Jersey, 1975.
8. Jury, E. I., Theory and Application of the z-Transform Method, John Wiley and Sons, Inc., New York, 1964.
9. Naval Air Development Center, Warminster, Pennsylvania, Report NADC-EL-47-50, Magnetic Airborne Detection Frequency Responses, by J. E. Anderson, 1949.
10. Stevens, Kurt B., Evaluation of the Naval Postgraduate School Geomagnetic Sensing System and Software, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1983.

BIBLIOGRAPHY

Bird, G. J. A., Design of Continuous and Digital Electronic Systems, McGraw-Hill Book Company Limited, 1980.

Close, C. M., The Analysis of Linear Circuits, Harcourt, Brace and World, Inc., 1966.

Humpherys, D. S., The Analysis, Design, and Synthesis of Electrical Filters, Prentice-Hall, Inc., 1970.

Jury, E. I., Theory and Application of the Z-Transform Method, John Wiley and Sons, Inc., 1964.

NAVAIR 16-30ASQ81-500, Preliminary Manual, Interim Field Engineering Maintenance Instructions with Illustrated Parts Breakdown, Magnetic Detecting Set AN/ASQ-81(V)2, (Texas Instruments, Inc.), 1 November 1971.

Oppenheim, A. V., and Schafer, R. W., Digital Signal Processing, Prentice-Hall, 1975.

Rabiner, L. R., and Gold, B., Theory and Application of Digital Signal Processing, Prentice-Hall, Inc., 1975.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, VA. 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, CA. 93943	2
3. Chairman ASW Academic Group, Code 71 Naval Postgraduate School Monterey, CA. 93943	1
4. Dr. Otto Heinz, Code 61Hz Department of Physics Naval Postgraduate School Monterey, CA. 93943	2
5. Dr. Andrew R. Ochadlick, Jr., Code 610c Department of Physics Naval Postgraduate School Monterey, CA. 93943	2
6. Dr. Paul Moose, Code 62Me Department of Electrical Engineering Naval Postgraduate School Monterey, CA. 93943	1
7. LT Michael C. Huete, USN Destroyer Squadron TWENTY-ONE FPO San Francisco, CA. 96601	2
8. Mr. John Shannon Naval Air Development Center (Code 3012) Warminster, PA. 18974	1
9. Mr. Edward Yannuzzi Naval Air Development Center (Code 30) Warminster, PA. 18974	1

10. Chief of Naval Research
Department of the Navy
Attn: Code 100C1 1
Code 425 GG 1
Code 460 1
Code 464 1
Code 480 1
800 North Quincy Street
Arlington, VA. 22217
11. Mr. William Andahazy 1
Naval Ship Research & Development Center
Annapolis Laboratory
Annapolis, MD. 21402
12. Naval Air Systems Command 1
Attn: Mr. Barry Dillon
Code AIR-340J
Washington, DC 20361
13. Fleet Tactical Library 1
P.O. Box 1042
Silver Springs, MD. 20910

202558

Thesis

H8535

Huete

c.1

A digital filter
representation of the
ASQ-81 magnetometer.

202558

Thesis

H8535

Huete

c.1

A digital filter
representation of the
ASQ-81 magnetometer.

thesH8535

A digital filter representation of the A



3 2768 002 13207 8

DUDLEY KNOX LIBRARY